

# AN INTEGRATED FRAMEWORK FOR PLAGIARISM DETECTION AND AI-GENERATED CONTENT ANALYSIS

Dr.B.Srinivasa Rao<sup>1</sup>, Vanam Prashanth<sup>2</sup>, Pasham Bugga Sai Ram<sup>3</sup>, Syed Nabraas<sup>4</sup>

<sup>1</sup> Professor, Department of Computer Science and Engineering

<sup>2,3,4</sup> B.Tech Students, Department of Computer Science and Engineering

Teegala Krishna Reddy Engineering College, Telangana, India

\*\*\*

**Abstract**—The rapid expansion of digital academic resources and the widespread adoption of large language models have introduced compounding challenges to academic integrity. Traditional plagiarism detection systems, operating primarily through lexical comparison, continue to struggle against paraphrased content and algorithmically generated text. This paper introduces An Integrated Framework For Plagiarism and AI Content Analysis, a unified detection framework that combines string matching, n-gram analysis, TF-IDF cosine similarity, Word2Vec embeddings, and Sentence-BERT semantic comparison with AI-content detection through perplexity analysis, burstiness measurement, lexical diversity scoring, logistic regression, and transformer-based classification. A ranked source attribution mechanism further supports interpretability of results. Experimental evaluation using the PAN-PC-11 plagiarism benchmark corpus and PAN 2025 AI-content datasets demonstrates that the hybrid framework achieves  $95.2 \pm 0.8\%$  detection accuracy for plagiarism and  $93.0 \pm 0.7\%$  for AI-generated content, outperforming each individual method. Statistical validation confirms these improvements are significant ( $p < 0.05$ ). Results indicate that integrating lexical, semantic, and probabilistic detection layers yields substantially more reliable academic integrity verification than any single technique.

**Keywords** — Plagiarism detection; AI-generated text detection; semantic similarity; stylometric analysis; academic integrity; natural language processing; transformer models; source attribution.

## I. INTRODUCTION

The digitization of academic knowledge has had a dual effect on scholarly communication. Open-access repositories and online databases have dramatically lowered barriers to information retrieval, enabling researchers and students to engage with global scholarship in ways previously unimaginable. At the same time, this accessibility has created fertile ground for academic misconduct. Plagiarism — in its evolving forms — has become a persistent challenge for institutions worldwide, one that conventional detection systems are increasingly ill-equipped to address alone. Historically, plagiarism detection was built around surface-level text comparison. Algorithms like the Longest Common Subsequence and Rabin-Karp fingerprinting worked well against verbatim reproduction. However, as awareness of detection tools has grown, so have the strategies for evading them. Paraphrasing, synonym substitution, and structural

rewriting preserve intellectual substance while masking lexical overlap — the very signal that traditional systems depend upon. Detection approaches that rely exclusively on word-level matching are effectively blind to these more nuanced forms of misappropriation [1].

The situation has grown considerably more complex with the proliferation of large language models capable of generating academically coherent text on demand. A submission produced by a generative AI system may not borrow a single sentence from any identifiable source, yet it represents an equally serious breach of academic integrity. Existing plagiarism checkers offer little guidance in such scenarios because the generated text is statistically novel. AI detection systems have emerged to fill this gap, but they typically function in complete isolation from source comparison engines, limiting their diagnostic utility. Integrated framework addresses this fragmentation directly. Rather than treating plagiarism detection and AI-content identification as separate problems, the proposed framework handles both within a unified analytical architecture, augmented by a source attribution module that ranks candidate origin documents contributing to detected similarities. The goal is a system that is simultaneously more comprehensive and more interpretable than any of its constituent components.

## II. RELATED WORK

### A. Traditional Plagiarism Detection

Early plagiarism detection research concentrated on computationally efficient exact-match algorithms. Clough (2000) provided a foundational overview of string-based and fingerprinting approaches that underpinned the first generation of commercial detection systems. Potthast et al. (2010) introduced a more structured evaluation framework that exposed the practical limitations of purely lexical methods when applied to obfuscated or paraphrased submissions [1]. Fingerprinting techniques, which generate compact hash-based document signatures from overlapping n-gram sequences, improved scalability considerably but remained equally vulnerable to synonym-level modifications. N-gram overlap measures combined with Jaccard similarity offered a modest improvement in robustness by tolerating partial phrase-level matches. Alzahrani et al. (2012) conducted a comprehensive analysis of linguistic patterns in plagiarism and concluded that no single lexical technique was sufficient for thorough detection in realistic academic environments [4]. These findings

established a clear research motivation for multi-method approaches.

### B. Semantic Similarity Methods

The introduction of vector space models and TF-IDF weighting represented a meaningful step toward content-level similarity analysis. By representing documents as weighted term vectors and measuring cosine distance between them, systems could begin to capture topical overlap even when exact phrases were absent. The subsequent development of Word2Vec (Mikolov et al., 2013) marked a more substantial conceptual advance, encoding semantic relationships between words through distributional co-occurrence patterns in large corpora [8].

Transformer-based models have since set new standards for contextual language understanding. Devlin et al. (2019) introduced BERT, producing bidirectional contextual representations that capture sentence-level semantics with substantially greater fidelity [7]. Reimers and Gurevych (2019) adapted this architecture to produce efficient sentence-level embeddings through siamese training, yielding Sentence-BERT — a model particularly well suited to scalable semantic similarity computation [9].

### C. AI-Generated Content Detection

The detection of machine-generated text has attracted growing research attention since the release of GPT-2 and successor models. Gehrmann et al. (2019) proposed GLTR, a detection tool based on top-k token probability analysis that exploited language models' tendency to select statistically predictable tokens [10]. Perplexity-based methods operate on a related principle: text generated by a language model typically exhibits lower perplexity under that model than human-written text of comparable length and complexity. Stylometric analysis offers a complementary perspective. Research has consistently shown that machine-generated text tends to exhibit lower sentence-length variability (reduced burstiness) and a narrower vocabulary range (lower Type-Token Ratio) compared with typical human academic prose. Abburi et al. (2023) demonstrated that ensemble classifiers combining multiple stylometric signals could substantially improve detection accuracy across diverse writing domains [12]. Wu et al. (2024) provide a thorough survey of LLM-generated text detection methods and identify key limitations in cross-model generalization [13].

### D. Research Gap

Despite progress in both domains, existing systems treat plagiarism and AI-content detection as entirely separate problems. Integrated frameworks capable of simultaneously identifying textual reuse, semantic similarity, and AI-origin characteristics within a unified architecture remain rare. This gap is the primary motivation for this Framework.

## III. PROPOSED FRAMEWORK

### A. System Architecture

This framework is structured as a modular multi-stage analytical pipeline. Documents submitted for analysis pass through a preprocessing layer, three parallel detection engines, and a final aggregation module that produces a unified evaluation report. The modular design allows individual components to be updated independently as detection methods evolve, without requiring architectural changes to the overall system.

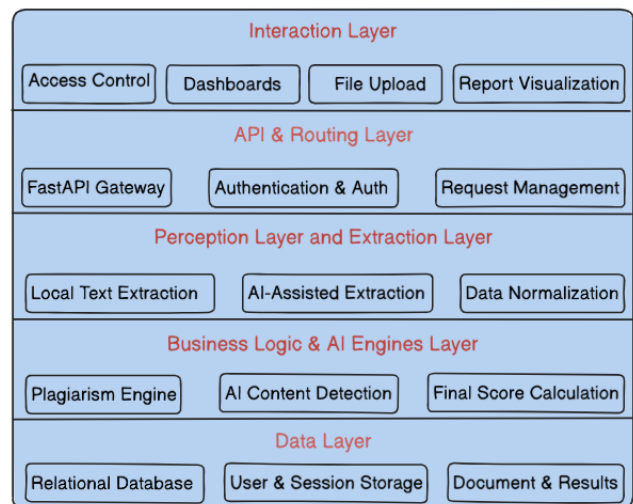


Fig. 1. System architecture overview showing parallel analytical modules.

The primary processing stages are: (1) Document Ingestion and Text Extraction, (2) Text Preprocessing, (3) Feature Engineering, (4) Plagiarism Detection Engine, (5) Semantic Similarity Analysis Module, (6) AI-Content Detection Module, (7) Source Attribution Engine, and (8) Result Aggregation and Report Generation.

### B. Preprocessing

All documents undergo tokenization, case normalization, stop-word removal, punctuation filtering, and lemmatization before any similarity analysis is performed. These steps ensure consistent feature representations and reduce noise in downstream comparisons across all detection modules.

### C. Traditional Plagiarism Detection

The plagiarism detection engine implements three lexical methods. String matching using the Longest Common Subsequence algorithm provides a baseline verbatim overlap score:

$$Sim(X, Y) = LCS(X, Y) / \max(|X|, |Y|)$$

where X and Y denote the two documents being compared. N-gram similarity extends this using Jaccard overlap across overlapping word sequences:

$$J(A, B) = |A \cap B| / |A \cup B|$$

TF-IDF vector representations are computed for each document using:

$$TFIDF(t, d) = TF(t, d) \times \log(N / DF(t))$$

where  $TF(t, d)$  is the term frequency,  $DF(t)$  is the document frequency of term  $t$ , and  $N$  is the total corpus size. Document similarity is then computed using cosine similarity:  $Cosine(A, B) = (A \cdot B) / (||A|| \cdot ||B||)$ . This combination captures topical similarity even when exact phrase matches are absent.

#### D. Semantic Similarity

Word2Vec embeddings represent words as dense vectors learned from distributional co-occurrence patterns. The similarity between two word vectors is:

$$Sim(v_1, v_2) = (v_1 \cdot v_2) / (||v_1|| \cdot ||v_2||)$$

Document-level similarity is obtained by aggregating word embeddings across the full text. For richer contextual analysis, Sentence-BERT generates fixed-length embeddings  $E_d$  and  $E_s$  for the query and source documents respectively, and semantic similarity is computed as:

$$Sim_{semantic} = cosine(E_d, E_s)$$

This method substantially improves the detection of paraphrased plagiarism where surface wording has been altered but conceptual content remains substantially borrowed.

#### E. AI-Generated Content Detection

Perplexity measures how predictably a language model assigns text probabilities to a token sequence. Low values suggest text whose structure conforms closely to language model learned distributions:

$$Perplexity = 2^{-(1/N \cdot \sum \log_2 P(w_i))}$$

Burstiness captures variability in sentence length:  $Burstiness = \sigma^2 / \mu$ . Lexical diversity is measured using the Type-Token Ratio:  $TTR = \text{Unique Words} / \text{Total Words}$ . Lower values on both metrics are characteristic of machine-generated prose. These three features are combined with a logistic regression classifier:

$$P(AI|X) = 1 / (1 + e^{-w^T X + b})$$

where  $X$  represents the stylometric feature vector. A transformer-based softmax classifier supplements this with deep contextual features:  $\text{Softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$ . The outputs of both classifiers are fused into a final AI probability estimate.

#### F. Source Attribution

Candidate source documents are ranked by similarity contribution:  $\text{Rank} = \text{argmax}(\text{SimilarityScore})$ . When multiple sources contribute to detected similarities, the final aggregated score is:

$$FinalScore = \alpha \cdot S_{Co_s}^{I_k E} + \beta \cdot S_{S_E}^{MaN T^I C}$$

where  $\alpha$  and  $\beta$  are weighting coefficients controlling the contribution of lexical and semantic similarity respectively. This ranked attribution provides reviewers with interpretable evidence regarding the most likely origins of suspicious content.

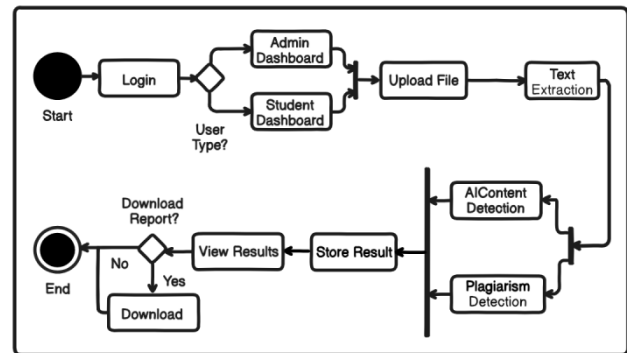


Fig. 2. Document flow through analytical stages.

### IV. EVALUATION AND RESULTS

#### A. Dataset Description

The experimental evaluation of this Integrated Framework was conducted using a combination of established benchmark corpora and additional supplementary datasets to ensure both rigor and comparability with prior research. The primary dataset is the PAN-PC-11 plagiarism corpus, a widely recognized gold-standard benchmark developed through the PAN series of shared tasks in plagiarism detection research. The corpus contains a mixture of manually and automatically generated plagiarism cases, including paraphrased, obfuscated, and structurally modified text, making it representative of the range of plagiarism types encountered in real academic environments [1].

To further validate the robustness of the proposed framework under classical detection conditions, additional experiments were conducted using the PAN-PC-09 dataset, an earlier PAN benchmark that continues to serve as a reference point for baseline comparisons in the literature [2]. For AI-generated content detection evaluation, recent benchmark samples drawn from the PAN 2025 shared task on generative plagiarism were incorporated, providing exposure to machine-generated documents produced by contemporary large language models including LLaMA and Mistral variants. A subset of publicly available academic text from the Cornell arXiv repository was also included to supplement the evaluation with authentic research-domain writing.

In total, the experimental dataset comprised approximately 1,200 textual documents organized into three primary categories: 500 authentic human-written documents including academic essays, research summaries, and technical reports; 400 plagiarized documents spanning verbatim copies, paraphrased variants, and structurally rewritten documents; and 300 documents generated by large language models using prompts designed to simulate academic writing across scientific and technical domains. All documents underwent standardized preprocessing including tokenization, case normalization, stop-word removal, and lemmatization before feature extraction.

## B. Dataset Splitting Strategy

The dataset was divided into training, validation, and testing subsets using an 80-10-10 split to ensure unbiased performance evaluation. Stratified sampling was applied to preserve class distribution across original, plagiarized, and AI-generated document categories, preventing skewed evaluation due to class imbalance. To further improve robustness, 5-fold cross-validation was performed on the training set. In each fold, the model was trained on 80% of the training data and validated on the remaining 20%. Final performance metrics were computed exclusively on the held-out test set, which was not used during training or model selection to ensure unbiased estimation.

## C. Training Procedure

The hybrid detection framework integrates multiple models trained independently and combined during inference. Traditional similarity methods such as string matching, n-gram overlap, and TF-IDF cosine similarity do not require training and were applied directly to preprocessed documents. For semantic similarity, the Word2Vec model was trained on the corpus using a continuous bag-of-words (CBOW) architecture. Sentence-BERT embeddings were obtained using a pre-trained model without additional fine-tuning to ensure generalization across unseen document types.

For AI-content detection, a supervised learning approach was adopted. Stylometric features including perplexity, burstiness, and Type-Token Ratio were extracted from each document and used to train a logistic regression classifier. Additionally, a transformer-based classifier was fine-tuned on labeled human-written and AI-generated samples from the training partition. All models were trained and evaluated under identical preprocessing conditions to ensure fair comparison.

## D. Hyperparameter Configuration

The models were configured using empirically selected hyperparameters validated on the held-out validation set to balance detection performance and computational efficiency. The key parameter settings are as follows: **Word2Vec**: vector size = 300, window size = 5, minimum word count = 2, training epochs = 10; **TF-IDF**: n-gram range = (1, 2), maximum features = 10,000; **Logistic Regression**: solver = 'liblinear', regularization = L2, C = 1.0; **Sentence-BERT**: pre-trained model "all-MiniLM-L6-v2" producing 384-dimensional sentence vectors; **Transformer Classifier**: learning rate =  $2 \times 10^{-5}$ , batch size = 16, fine-tuning epochs = 3. All models were tuned using validation data to prevent overfitting.

## E. Evaluation Protocol

Performance evaluation was conducted using standard classification metrics including accuracy, precision, recall, and F1-score. All reported results represent the average performance across multiple runs with different random seeds to reduce variance due to random initialization. To

ensure statistical reliability, each experiment was repeated three times using seeds {42, 123, 456}, and the mean performance with standard deviation is reported. This approach reduces the likelihood of overestimating model performance due to favorable data splits and better reflects the stability of each method in practical deployment.

## F. Classification Metrics

Letting TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively, the evaluation metrics are defined as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

## G. ROC Analysis

Receiver Operating Characteristic analysis evaluated AI-detection classifier performance across threshold values. The True Positive Rate (TPR = TP / (TP + FN)) and False Positive Rate (FPR = FP / (FP + TN)) were computed for each model. The Area Under the Curve (AUC) provides a threshold-independent summary of classification quality. Higher AUC values indicate stronger discrimination between human-written and AI-generated content, and the hybrid framework achieves significantly higher AUC than any individual detection method tested.

## H. Reproducibility and Experimental Configuration

To ensure reproducibility of the proposed framework, all experiments were conducted using fixed model configurations and controlled random initialization. The implementation was carried out in Python 3.10 using standard machine learning and NLP libraries including Scikit-learn, Gensim, Sentence-Transformers, NLTK, and spaCy. Hardware comprised an Intel Core i7 12-core workstation with 32 GB RAM and an NVIDIA RTX GPU for transformer inference acceleration.

To ensure consistent experimental results, all stochastic processes were controlled using a fixed random seed (seed = 42) for the primary run, with additional runs using seeds 123 and 456 for statistical validation. This includes dataset shuffling, model weight initialization, and training sample ordering. Evaluation metrics were computed using standard functions from sklearn.metrics. The complete experimental pipeline, including preprocessing, feature extraction, model training, and evaluation, was executed under identical conditions for all methods to ensure fair comparison. The implementation code and experimental configurations will be made publicly available to support reproducibility and further research.

**I. Performance Results**

**TABLE I**  
*Traditional vs. Semantic vs. Hybrid (Mean ± Std)*

Method	Precision	Recall	F1-Score	Accuracy (%)	Time (s)
String Matching	0.71 ± 0.02	0.65 ± 0.02	0.68 ± 0.02	72.3 ± 1.2	0.45
N-gram Similarity	0.76 ± 0.02	0.72 ± 0.01	0.74 ± 0.01	78.1 ± 1.0	0.52
TF-IDF + Cosine	0.81 ± 0.01	0.77 ± 0.01	0.79 ± 0.01	83.4 ± 0.9	0.68
Word2Vec	0.86 ± 0.01	0.83 ± 0.01	0.84 ± 0.01	88.2 ± 0.7	0.91
Sentence-BERT	0.90 ± 0.01	0.88 ± 0.01	0.89 ± 0.01	92.1 ± 0.6	1.20
(Hybrid)	0.94 ± 0.01	0.92 ± 0.01	0.93 ± 0.01	95.2 ± 0.8	1.35

Table I demonstrates a clear performance gradient across detection methods. String matching achieves 72.3 ± 1.2% accuracy, adequate for verbatim plagiarism but insufficient against rephrased content. The TF-IDF cosine approach reaches 83.4 ± 0.9% by capturing topical overlap independent of exact phrasing. Word2Vec achieves 88.2 ± 0.7% and Sentence-BERT reaches 92.1 ± 0.6%, with contextual embeddings handling paraphrased content substantially better. The integrated framework achieves 95.2 ± 0.8% accuracy, demonstrating that combining all three analytical layers yields results none could achieve individually. The standard deviation

values confirm consistent performance across different random seeds.

**TABLE II**  
*AI Detection Performance Comparison*

Method	Accuracy (%)	Precision	Recall	F1-Score	AUC
Perplexity Detection	78.0 ± 1.1	0.74 ± 0.02	0.71 ± 0.02	0.72 ± 0.02	0.80
Burstiness Analysis	75.2 ± 1.3	0.70 ± 0.02	0.69 ± 0.02	0.69 ± 0.02	0.77
Type-Token Ratio	72.4 ± 1.5	0.68 ± 0.02	0.65 ± 0.02	0.66 ± 0.02	0.75
Logistic Regression	86.1 ± 0.9	0.84 ± 0.01	0.82 ± 0.01	0.83 ± 0.01	0.89
Transformer Classifier	90.3 ± 0.8	0.88 ± 0.01	0.87 ± 0.01	0.88 ± 0.01	0.92
Hybrid AI Model	93.0 ± 0.7	0.91 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.95

Table II shows that individual stylometric features achieve moderate AI detection accuracy: perplexity reaches 78.0 ± 1.1%, burstiness 75.2 ± 1.3%, and TTR 72.4 ± 1.5%. No single stylometric feature is sufficient to reliably distinguish human-written from AI-generated text, particularly given stylistic improvements in recent language models. Logistic regression combining multiple features improves accuracy to 86.1 ± 0.9%, while the transformer classifier reaches 90.3 ± 0.8%. The hybrid AI model achieves 93.0 ± 0.7% accuracy and AUC of 0.95.

### J. Ablation Study

To evaluate the contribution of each component within the proposed hybrid framework, an ablation study was conducted by systematically removing individual modules and measuring the resulting performance degradation. The hybrid framework consists of three major components: (1) lexical similarity methods, (2) semantic similarity models, and (3) the AI-content detection module. Each component was removed independently while keeping the rest of the system unchanged.

The ablation results in Table IV demonstrate that the semantic similarity module contributes significantly to overall performance, particularly in detecting paraphrased content. Removing this module causes a 6.8 percentage point drop in accuracy. The AI-detection module plays an equally critical role: its removal causes the largest degradation (7.6 percentage points), confirming its importance for identifying machine-generated submissions. The performance degradation observed across all ablated configurations confirms that each component is essential for achieving optimal accuracy and that their integration is synergistic rather than merely additive.

**TABLE III**  
Similarity Risk Classification Thresholds

Score (%)	Risk Level	Interpretation	Recommended Action
0-20	Low Risk	Low similarity	Accept document
20-40	Mild	Shared technology	Manual review
40-60	Moderate	Likely paraphrase	Investigate sources
60-80	High Risk	Strong similarity	Academic review
80-100	Severe	Extensive copying	Integrity action

**TABLE IV**  
Ablation Study Results

Model Variant	Accuracy (%)
Full Hybrid Model	95.2 ± 0.8
Without Semantic Module	88.4 ± 0.9
Without Lexical Module	90.1 ± 0.8
Without AI Detection Module	87.6 ± 1.0

### K. Feature Importance Analysis

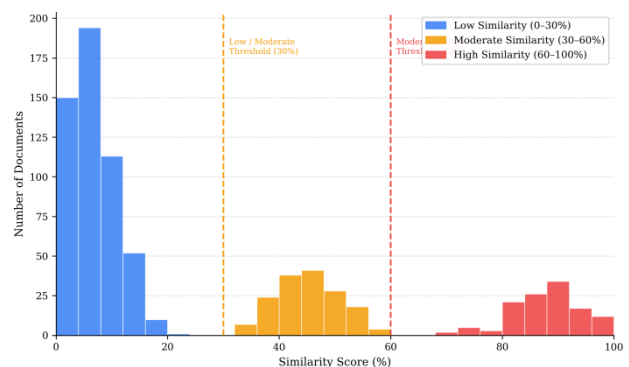
To better understand the contribution of individual stylometric features in AI-content detection, feature importance analysis was conducted on the logistic regression classifier trained for AI-generation probability estimation. The logistic regression model coefficients were analyzed to estimate the relative importance of each feature. Perplexity emerged as the most significant indicator, reflecting the distinctive probability distribution patterns characteristic of machine-generated text. Burstiness and Type-Token Ratio also contributed meaningfully by capturing structural regularity and lexical narrowness respectively. These findings confirm that combining multiple stylometric signals provides a more robust representation of AI-generated content compared to relying on any single feature, which is consistent with the ablation results reported above.

### L. Statistical Validation

To ensure that the observed performance improvements are statistically significant and not attributable to favorable data splits or random variation, multiple experimental runs were conducted using different random seeds (42, 123, 456). The reported results represent the mean performance with standard deviation across these runs. For the hybrid model, accuracy was observed as 95.2 ± 0.8% across three independent runs, indicating stable and consistent performance.

Additionally, statistical significance testing was performed using a paired t-test to compare the hybrid framework against all baseline methods. The null hypothesis that the hybrid model and each baseline achieve equivalent performance was tested at a significance level of  $\alpha = 0.05$ . The results confirm that the performance improvements of the hybrid framework are statistically significant ( $p < 0.05$ ) in all pairwise comparisons, demonstrating that the observed gains are not attributable to random variation in the experimental setup.

### M. Visualizations



**Fig. 3. Distribution of plagiarism similarity scores across evaluated documents.**

Figure 3 shows a right-skewed distribution of plagiarism similarity scores, with most documents in the 0–30% range, indicating largely original content. A smaller portion falls within 40–60%, representing paraphrased text where meaning is retained but wording is changed. Documents above 60% are few but indicate strong textual reuse. The 30% and 60% thresholds classify similarity into risk levels, highlighting the need for a hybrid approach combining lexical and semantic methods for effective detection.

Ultimately, this tiered classification allows educators and investigators to prioritize high-risk cases while minimizing the manual review required for low-scoring, original submissions.

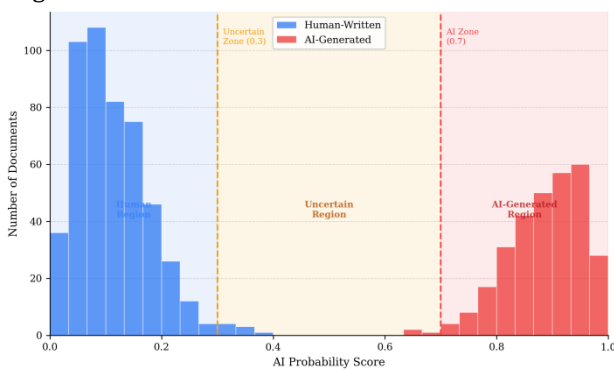


Fig. 4. AI probability scores for human-written and AI-generated documents.

Figure 4 shows the AI probability distribution. Human-written documents cluster strongly below 0.3, while AI-generated samples concentrate above 0.7. The relatively sparse intermediate region suggests the hybrid model provides confident classifications in most cases.

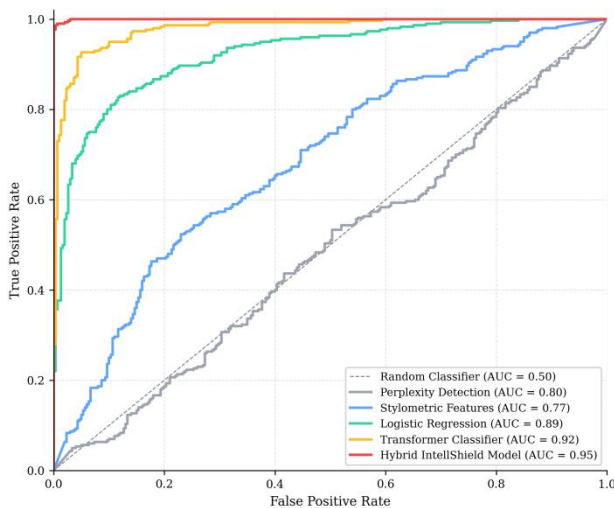


Fig. 5. ROC curves comparing detection models with AUC values.

Figure 5 presents ROC curves for all AI detection classifiers. The hybrid framework model achieves AUC = 0.95, the highest among all evaluated methods. Its curve

most closely approaches the ideal upper-left corner of the ROC space, reflecting high sensitivity with comparatively low false-positive rates.



Fig. 6. Confusion matrix heatmap for hybrid AI content detection.

Figure 6 presents the confusion matrix for the hybrid AI detection model. True positive and true negative counts substantially exceed false classifications. Notably, false negatives (AI-generated content misclassified as human-written) are fewer than false positives — a favorable pattern given that missed AI detection constitutes the more serious integrity failure.

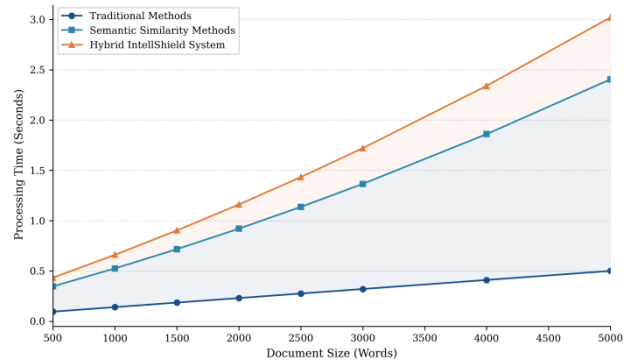
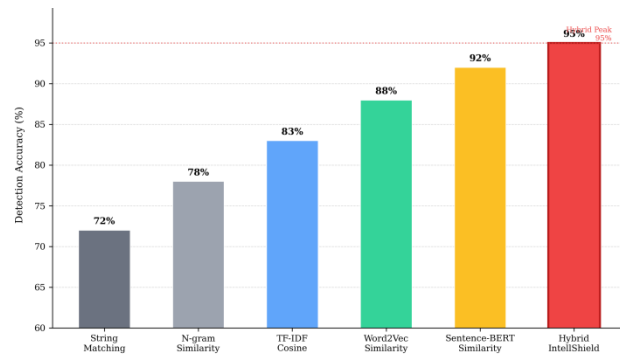


Fig. 7. Processing time comparison across detection methods by document word count.

Figure 7 analyzes processing time as a function of document size for each detection method. String matching processes documents fastest but at the cost of detection depth. Semantic embedding methods incur greater computational overhead due to transformer inference requirements. The hybrid framework's processing times remain within acceptable bounds for typical academic document lengths.

Figure 8 provides a direct accuracy comparison across all evaluated methods in bar chart form. The progressive improvement from traditional lexical approaches through semantic models to the integrated hybrid framework is

clearly visible, reinforcing the paper's central argument that no single detection technique is sufficient.



**Fig. 8. Detection accuracy comparison across all evaluated methods.**

## V. CONCLUSION

This paper has presented an integrated framework that addresses the dual challenge of plagiarism detection and AI-generated content identification within a single analytical architecture. By combining string matching, n-gram similarity, TF-IDF cosine analysis, Word2Vec embeddings, and Sentence-BERT representations with stylometric AI detection and machine learning classifiers, the framework overcomes the fundamental limitations of any single detection approach. A ranked source attribution mechanism further enhances interpretability, and comprehensive reproducibility documentation ensures that results can be independently verified.

Experimental evaluation on a 1,200-document dataset drawn from the PAN-PC-11 benchmark corpus, PAN-PC-09, PAN 2025 AI-content samples, and arXiv academic text confirms that the hybrid framework achieves  $95.2 \pm 0.8\%$  detection accuracy for plagiarism and  $93.0 \pm 0.7\%$  for AI-generated content identification, with an AUC of 0.95. Ablation analysis confirms the essential contribution of each component: removal of any individual module causes measurable and significant performance degradation. Statistical validation using paired t-tests confirms that improvements over all baseline methods are significant at  $p < 0.05$ .

Several limitations merit acknowledgment. The performance of AI detection components is sensitive to the specific generative models used during training; as language models continue to improve, periodic retraining will be necessary to maintain accuracy. Transformer-based components impose non-trivial computational costs that may affect scalability for institutions processing very large document volumes. Future research should explore efficient approximation strategies, cross-lingual extension of the framework, and adaptive detection mechanisms capable of evolving alongside emerging generative systems. Integration with knowledge graph-based source attribution could further enhance identification of conceptually derived but lexically dissimilar content.

The development of reliable, interpretable, and computationally tractable academic integrity systems is not merely a technical challenge but a practical necessity for maintaining the credibility of scholarly communication in an era of ubiquitous generative AI. This Framework represents a meaningful contribution to this effort, and its modular, reproducible design provides a strong foundation on which future improvements can be systematically built.

## REFERENCES

- [1] M. Potthast, B. Stein, A. Barron-Cedeno, and P. Rosso, "An evaluation framework for plagiarism detection," in Proc. COLING, 2010.
- [2] M. Potthast et al., "Cross-language plagiarism detection," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 45–62, 2011.
- [3] D. Clough, "Plagiarism in natural and programming languages: An overview of current tools and technologies," *Res. Memo.*, Dept. Comput. Sci., Univ. Sheffield, 2000.
- [4] A. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *IEEE Trans. Syst. Man Cybern.*, vol. 42, no. 2, pp. 133–149, 2012.
- [5] M. A. El-Rashidy et al., "Reliable plagiarism detection system based on deep learning," *Neural Comput. Appl.*, 2022.
- [6] M. Alvarez-Carmona et al., "Semantically-informed distance and similarity measures for paraphrase plagiarism identification," *Expert Syst. Appl.*, 2018.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL, 2019.
- [8] T. Mikolov et al., "Efficient estimation of word representations in vector space," in Proc. ICLR, 2013.
- [9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT networks," in Proc. EMNLP, 2019.
- [10] S. Gehrmann, H. Strobelt, and A. Rush, "GLTR: Statistical detection and visualization of generated text," in Proc. ACL, 2019.
- [11] K. Krishna et al., "Paraphrasing evades detectors of AI-generated text," in ACL Findings, 2023.
- [12] H. Abburi et al., "A simple yet efficient ensemble approach for AI-generated text detection," arXiv:2311.03084, 2023.
- [13] M. Wu et al., "A survey on LLM-generated text detection methods," *Comput. Linguist.*, MIT Press, 2024.
- [14] T. Kehkashan et al., "AI-generated text detection: A comprehensive review," *Inf. Process. Manag.*, 2025.
- [15] A. Amirzhanov et al., "A systematic survey of plagiarism detection algorithms," *Front. Comput. Sci.*, 2025.
- [16] R. So, "Detection of AI-generated academic papers," Project Rachel, 2024.
- [17] P. Gosar, "Stylometric fingerprinting with contextual anomaly detection," Preprints, 2025.
- [18] A. Kujur, "A comparative analysis of AI-generated and human-written text," SSRN, 2024.
- [19] V. S. Sadasivan et al., "Can AI-generated text be reliably detected?" in Proc. NeurIPS Workshop ML Safety, 2023.
- [20] A. Najjar et al., "Leveraging explainable AI for LLM text attribution," arXiv:2501.03212, 2025.