scientific reports

Check for updates

OPEN

Identifying artificial intelligence-generated content using the DistilBERT transformer and NLP techniques

Hikmat Ullah Khan[1]✉, Anam Naz[1], Fawaz Khaled Alarfaj[2]✉ & Naif Almusallam[2]

Natural language processing (NLP) has evolved significantly with the emergence of large language models (LLMs), leading to the rapid growth of artificial intelligence-generated content (AIGC). This expansion raises critical challenges in ensuring content authenticity and preventing the spread of misinformation and plagiarism. The identification of AIGC is an active research area and is significant for maintaining the authenticity and credibility of digital content in academic and professional environments. In this research study, we focus on identifying AIGC based on state-of-the-art deep learning-based transformers and by exploring deep features of textual content. The proposed DistilBERT transformer is an advanced and lightweight form of bidirectional encoder representations from transformers (BERT), a well-known transformer of many LLMs that utilizes a distilled transformer architecture with self-attention mechanisms that dynamically weigh textual elements based on contextual relevance, effectively capturing linguistic patterns. Additionally, this study explores both traditional machine learning with textual features and deep learning models integrated with word embeddings such as GloVe and Fast Text. Experimental analysis demonstrated that the proposed DistilBERT-based model achieved a superior predictive accuracy of 98%, outperforming traditional deep learning models, such as long short-term memory (LSTM) with GloVe embeddings, which achieved 93% accuracy. Furthermore, qualitative assessments validate the model's effectiveness in confidently classifying diverse textual samples, reinforcing its practical reliability.

Keywords Artificial intelligence, Natural language processing, AI generation, Deep learning, Academics, Content verification, Text classification

Artificial intelligence (AI) has already revolutionized our lives owing to its applications in every aspect of human life. AI-generated content is transforming numerous sectors by automating the production of text, images, music, and videos. This technology is being used by everyone to improve their work and make it more efficient. It utilizes sophisticated AI-based algorithms and machine learning models to generate high-quality content quickly and at large scales[1]. AI-generated content helps us to save time and resources while maintaining creativity and personalization. For example, in the business sector, AI-generated content (AIGC) can streamline marketing efforts, improve customer engagement, and reduce operational costs[2]. Thus, the rise of AI tools such as LLMs and ChatGPT has already affected every business[3]. Furthermore, it opens new creative avenues, enabling individuals and organizations to explore innovative forms of expression[4].

The identification of AI-generated content (AIGC) is increasingly recognized as a significant research challenge for various reasons, such as trust in media, credibility of content and issues of academic integrity[5]. As generative AI models become more sophisticated, distinguishing between human-written and machine-generated text is crucial in various contexts, particularly in education and journalism[6]. For instance, educators are compelled to utilize AIGC detectors to ascertain whether students are engaging in academic dishonesty, as reliance on AI-generated submissions can undermine the learning process[7]. While existing AIGC detectors have shown proficiency in identifying AI-generated text, their effectiveness in recognizing AI-generated code remains uncertain due to the complex nature of programming languages[8]. This discrepancy can lead to disparities in the evaluation of students' academic submissions, potentially resulting in unfair trading practices[9].

[1]Department of Information Technology, University of Sargodha, Punjab, Pakistan. [2]Department of Management Information Systems, School of Business, King Faisal University, Al Ahsa, Saudi Arabia. ✉email: dr.hikmat.niazi@gmail.com; falarfaj@kfu.edu.sa

The growing reliance on AIGC raises ethical concerns about authorship and originality. Many academic journals have stringent policies regarding the use of AI-generated content, requiring authors to disclose their use of AI tools and methodologies. Failure to do so can lead to severe consequences, including damage to the author's reputation and rejection of manuscripts[10]. Moreover, the limitations of current detection tools highlight the need for further research into improving the accuracy and generalizability of these tools across different content types and programming languages[11]. Studies have shown that existing AIGC detectors often perform poorly when applied to source code generated by various generative models, indicating a significant gap that needs to be addressed. It is, therefore, important to note that the contribution of this study resides in the attempt to address emerging issues emanating from the rising use of AI-generated content in media[12]. It has become paramount in addressing questions about the misinformation, quality, and originality of text and the nature of communication as AI technologies develop. The rationale for this research comes from the concern of finding scalable and effective models to detect AI-created content to combat potential AI-enabled misinformation[13]. The detection approach used in this research incorporates deep learning and additive methods to offer a sound method that can improve the ability to identify manipulation while acting as a basis for future developments in counteracting the manipulation of new media[14]. Despite substantial progress in detecting AI-generated text using transformer architectures, existing approaches often overlook the combined use of subword-level and contextual embeddings alongside traditional linguistic features, as well as the systematic comparison of shallow, recurrent, and transformer-based models under identical preprocessing and evaluation settings. In contrast, this study introduces three key novelties: (1) integration of advanced dense embeddings—FastText's subword vectors alongside GloVe to capture both semantic and morphological nuances; (2) comprehensive benchmarking of classical (TF-IDF, N-gram, POS), deep (RNN, LSTM), and large-language-model representations within a unified framework; and (3) exploitation of DistilBERT's multi-head self-attention blocks, integrating pretrained contextual embeddings to dynamically weight stylistic cues that distinguish AI from human authorship. By conducting extensive experiments and rigorous statistical testing based on model confidence analysis on each feature-model combination, this work delivers the most comprehensive and methodological improvement in comparison to detect text-nature. Finally, the research provides a methodological contribution toward addressing transparency and accountability demands at a time when the distinction between who is writing what using AI becomes blurry.

In this research study, our aim is to detect AI-generated text content from humans on a textual dataset that is publicly available in the Kaggle repository with 500 k essays in a dataset created by both AI and humans. For this analysis, the dataset was split into 80–20 subsets by exploring the role of various textural features used as inputs to machine learning classifiers and with ensemble learning algorithms. In addition, various deep features that focus on considering context for given words from local and global perspectives are also applied in comparison with state-of-the-art transformer-based models. The results are evaluated using standard performance metrics, namely, accuracy, precision, recall and F-measure.

The main contributions to this research include the following:

- A state-of-the-art transformer-based model (DistilBERT) is applied, and its attention mechanism is used to effectively capture global dependencies and significant contextual patterns in textual data for accurate authenticity detection.
- The highest identifying accuracy of 98% was achieved, demonstrating superior performance over traditional machine learning and deep learning models.
- We conducted a comprehensive comparative analysis using traditional machine learning models with textual features and deep learning models with word embeddings.
- A detailed performance evaluation was performed through validation accuracy analysis across epochs, qualitative analysis of individual predictions, and model confidence assessment to ensure the robustness and reliability of the proposed method.

The remainder of the paper is structured as follows and illustrated in Fig. 1: Section "Related work" defines the background using existing studies based on different computational models. Section "Research proposed methodology" shares the in-depth analysis of the proposed methodology used to conduct this study. Section "Results and discussion" provides the results and discussion obtained from the applied methodology. Section "Conclusion and future work" presents a summary of this work and discusses future research directions.

Related work

In this section, we review the relevant articles that consider machine learning and deep learning for identifying AIGC from textual content. Machine learning models have been widely used for classifying humans and AIGC for text data. To accomplish this task, recent studies share the use of conventional machine learning algorithms such as decision trees[15], which have achieved excellent results using stylometric features, which usually share details about n-grams, the positioning of commas and the frequency of use of function words. In addition to shallow machine learning algorithms, the application of ensemble models such as XGBoost[16] has shown promising results in terms of student essays. The model achieved excellent results, with up to 98% accuracy, when features such as TF-IDF and other handcrafted features were used.

State-of-the-art deep learning-based models have also been used for identifying text AIGC. The transformer classifier has been used on a dataset containing customer reviews, and it achieved an accuracy of 79%[17]. The variant of BERT, DistilBERT, has also been used for identification of AIGC- vs. human-generated content, with promising results of 90%[18]. For AIGC detection, various approaches, ranging from conventional content-related approaches to state-of-the-art deep learning transformer-based approaches, have been identified. Research by Tien and Labbe[19] used short textual fragments and considered a parse tree for AIGC. Then, the fragments

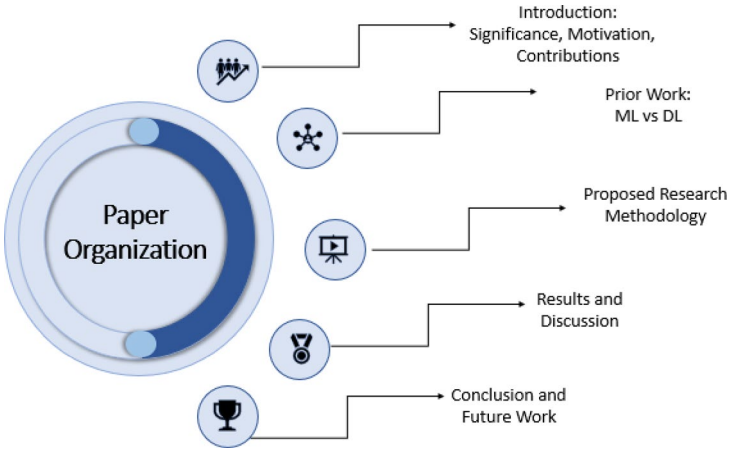

Fig. 1. Organization of the Paper.

and parse tree structure are used in diverse types of algorithms, such as Markov models and recurrent neural networks. The model achieved an 80% positive detection rate and a false detection rate of less than 1%. Another study[20] integrated Markov models with probabilistic content free grammar (PCFG) by considering factors such as sentence length, sentence structure, vocabulary richness and word usage count. However, AI lacks the ability to diversify vocabulary across diverse scenarios.

Social media content generation using AI tools is common, and fake news and disinformation are also spread due to this. Disinformation is the intended alteration in content with the intention to deceive someone[21]. Misinformation, on the other hand, is spread of unauthentic information without negative intentions, and fake news is created to mislead the public. Stiff and Johansson[22] studied the role of transformer-based models using news articles and social media posts. The authors concluded that the latest deep learning algorithms fail to detect short social media items that consist of few words, one sentence or two words. AIGC is being used for academic writing, and the detection of such content is important for maintaining the integrity of professionals working in this area. In this regard, another study[23] used automatic article generator tools that use keywords and article topics from users and generated articles using natural language content based on the linkage of concepts via a knowledge tree. In the future, academics will face serious challenges in understanding the true potential of students and researchers. Furthermore, another study[24] focused on plagiarism detection in the content generated by AI tools. They explored various techniques and reviewed many existing studies and determined that with advancements in AI tools for content generation, there is a need to enhance plagiarism detection techniques as well.

Focusing on feature-centric approaches, Akbari and Arabi[25] worked on identifying extrinsic plagiarism through a two-stage study. First, they focused on textual features such as bags of words, in which each word is considered independent of other words and not considering its sequence with other words. They carried out this analysis at the sentence and document levels. Then, they used WordNet ontology, which is the world standard ontology that helps us to obtain the synonym of the word; thus, paraphrasing can be detected. Another study[26] involved the use of various plagiarism tools, and they provided comparative analysis of existing plagiarism tools such as Turnitin and Dupli Check. Another study by Khalil and Er[27] also involved AIGC-based plagiarism detection, and they concluded that the common means of new automatic content generation in academic writing are based on paraphrasing, repetition of content and verbatim writing.

Another study[28] addressed how to leverage linguistic features automatically via computational tools to discriminate human-authored and AI-generated text. The tool provided interpretability and transparency, with an accuracy of 87.2%. However, the model is vulnerable to clergy mimicry by fresh LLMs, and the lack of deep learning integration may hinder scalability across many contexts. Moreover, deep learning-based model in comparison of machine learning[29] used support vector machines (SVMs) and a random forest model on a balanced dataset of 3000 samples (1500 human and artificial intelligence (AI)). This study made a clean mark in establishing a benchmark for classical methods with 92.4% accuracy. This is because there is no comparison with transformer-based architectures, which has been used for recent NLP tasks. In their work, Kumar et al.[30] used smaller semantic differences by constructing ML classifiers from word embeddings. The study is strong, as it uses detailed comparative lens and linguistic analysis. However, the accuracy of 85.3% is relatively low and might indicate performance improvement in exchange for interpretability. Furthermore, generative model styles may evolve, and the models may not adapt well to them. In addition, Twiari et al.[31] proposed a hybrid architecture that combines shallow and deep learning approaches for identifying GPT-generated news articles. The practical utility and high performance (88.7%) of the framework were confirmed. The evaluation was restricted to news data, and there was no cross-domain generalizability. Additionally, neither ethical benchmarking nor adversarial testing was performed.

According to Sardinha[32], he used a corpus linguistics approach to compare texts of differing registers on different dimensions. The results of the study showed that genre and register play a large role in the interpretability of AI content when language is contextual. The scope of the study was novel, but the accuracy of the model

(81.9%) was low, and the lack of algorithmic diversity made the method less broadly applicable. Boutadjine et al.[33] also made a large leap by fine tuning a RoBERTa model on human vs. machine-generated texts, obtaining 93.6% accuracy. The purpose of this study was to incorporate both qualitative and quantitative metrics. Just as important, this scale also assessed generalization across domains. However, the limitation comes from the low transparency around which the dataset is composed, which makes reproducibility and fairness assessment difficult. A new benchmark dataset was introduced by Al Bataineh et al.[34] to evaluate AI-human text detection comprehensively. The researchers used BERT-based models, which reached up to 90.8% accuracy, and stressed the importance of ethical AI evaluation. This paper is impressive, both because it is highly data centric and because there is real-world benchmarking. Nevertheless, adversarial examples or multilingual testing could increase the scope since the study is primarily monolingual and does not include stress tests for model robustness.

Furthermore, recent work has focused on differentiating AI and human text but is challenged by powerful language models and new adversarial techniques. A study by[35] experimenting with the resistance of AI detection on adversarial modified AI generated text and prompting detectors to evolve with evasion strategies. Moreover, multilingual AI-generated essay detection, focusing on the necessity of cross-lingual skills in detection models[36]. In comparison with state-of-the-art recognition systems, GPTZero, Turnitin, Perplexity, Grok, and DetectGPT, and other open-source detectors were compared by[37] which unveiled them being effective against adversarial prompts. Their results reveal that some tools exhibit strong robustness while others are vulnerable to evasion attacks. In addition, a new detection framework was proposed in[38] which dynamic perturbations are employed for better generalization ability and robustness across various text domains. To tackle these issues with adversarial prompt generation[39] introduced a token probability-based method by exploiting the embeddings for constructing adversarial attacks and observed that current detection models are defenseless. In response,[40] proposed adopting a defense technique based on diffusion learning that intends to enhance the model robustness against these types of attack. Cross-lingual and cross-modal detection is an important research for detection of AI-g artificial content, showing that multi-lingual models tend to break on more languages diversity datasets[41] to take various linguistic features to achieve high detection accuracy in multilingual environments.

The existing studies in AI-generated text detection have concentrated on the balanced in-domain benchmarks and mainly monolingual corpora, rely on static linguistic features (TF-IDF, n-grams, POS tags) or off-the-shelf transformer embeddings without any targeted adaptation, resulting in brittle performance when faced with domain shifts, mixed-authorship documents, or short, interleaved AI fragments. Furthermore, very few provide a proper statistical validation or imbalance-aware metrics, so results might be overly broad and unrealistic, and nearly none tackle both adversarial evasion strategies, cross-lingual validity, and multimodality. This creates a significant barrier to deploying strong, scalable detectors that are equipped to deal with the covert, adversarial, and multilingual character of today's AI-supported communication. The review of existing literature in the relevant field, defined in Table 1, suggests that current approaches to detect AI-generated text content has yet to achieve high accuracy but it lacks to keep the pace due to speed of its diversity and complexity of natural language content. Therefore, there is a need to carry out continued research to address the challenging task and ensure the reliability and integrity of information whether it is in social media or in academic articles.

Research proposed methodology

The following section discusses the approaches used to process the data while retrieving relevant features for subsequent analysis and constructing the models. The steps of the methodology include data preprocessing, feature extraction and applying the model to the data with the feature space; this is another step in determining the experimental outcomes, following the methodology shown in Fig. 2. This figure shows that the pipeline selection of each feature extraction and modeling tool is suited for different aspects of this study, which is why they were both considered. TF-IDF allows for a basic interpretation by identifying the most significant terms, and this process can be applied on its own. With GloVe, models can benefit from understanding connections between words that are based on the way they are commonly used in a large set of texts. With FastText, the level of detail is raised by considering short subword units, helping solve some issues that designers of language models often face. Lastly, DistilBERT, which is a small version of transformers, gives the best contextual embeddings and uses multiple self-attention blocks trained to notice key messages among all the words, unlike other models. This methodology goes beyond merely combining established tools by introducing targeted adaptations—namely, a class-weighted loss function to address the inherent human–AI class imbalance and a tailored self-attention head that emphasizes stylistic and structural cues unique to AI-generated text. Compared to similar transformer-based detectors, this method benefits from dynamic token-level augmentation during fine-tuning—enhancing robustness to noise and domain shifts—and from rigorous statistical validation that confirms its gains are not due to chance. The innovation lies in the tight integration of imbalance-aware optimization, multi-granularity embeddings, and targeted attention adaptation, yielding a detector that is both more accurate (98% accuracy, 98% F1) and more reliable across diverse text domains. Overall, the tools allow for a detailed assessment using different AI approaches to help discover the best approach for recognizing if text comes from AI or a human.

Data preprocessing

The process involves activities to ensure that the input data are clean, consistent, and ready for analysis. The steps used for data preprocessing are as follows:

Stopword removal: Stop words, which are defined as words that do not add to the meaning of the document when searched for, were eliminated, such as "the", "and", among others.
Digit, Punctuation, Link, and Special Character Removal: This approach to some extent minimized confusion by removing nonalphanumeric characters, links, and icons that are not informative, as defined in Eq. (1).

References	Model	Dataset	Feature	Results	Strength	Limitation
[19]	RNN, Markov model	Reddit, Yahoo Answers	N-Gram, TF-IDF	88	Combines sequential and probabilistic models for robust pattern detection	Limited contextual understanding; struggles with long-range dependencies
[20]	SVM, KNN, Decision Tree	Research paper content	POS tagging	85	Leverages syntactic cues via POS features for clear interpretability	Classical classifiers may overfit on limited syntactic patterns; poor semantic generalization
[24]	CNN, RNN	Research paper content	N-Gram, POS	85	Utilizes convolutional layers for local pattern extraction and recurrent memory for context	High computational cost; sensitivity to hyperparameter tuning
[35]	Naïve Bayes, LSTM	WordNet and PAN human-written texts	Textual feature sets	90	Combines probabilistic baseline with deep sequence modeling for balanced performance	Naïve Bayes overly simplistic; LSTM requires extensive training data
[22]	RoBERTa	Yelp user reviews	Default transformer encoding	91	State-of-the-art contextual embeddings capture nuanced sentiment and style	Large model size leads to high inference latency and resource demands
[25]	RoBERTa WordNet ontology	Tweets, Reddit comments, Yahoo answers, and Yelp user reviews. weets, Reddit comments, Yahoo answers, and Yelp user reviews. tweets, Reddit comments	Default feature encoding	91	Subword-level embeddings handle misspellings and rare words effectively	Ontology reliance may introduce bias; limited to covered semantic relations
[17]	BERT	Essays	TF-IDF	79	Fine-tuned transformer demonstrates baseline applicability to structured essay texts	TF-IDF lacks semantic depth; model underperforms on free-form or noisy inputs
[18]	BERT	Essays	Default feature encoding	66	Leverages pretrained contextual knowledge	Low accuracy indicates overfitting to training domain; limited feature adaptation
[27]	GRU	Essays, Tweets, Yelp	Count vectorization	87	Gated units capture sequence dynamics with moderate resource usage	Simpler than LSTM; may miss very long-term dependencies
[16]	SVM, Logistic Regression, RF, DT	BBC News	Textual features	89	Comprehensive comparison of multiple shallow classifiers highlights best performer	Shallow methods struggle with semantic nuances; inconsistent performance across topics
[28]	SVM, GBM, DT	Online text corpus (AI vs. Human)	Linguistic feature fusion	87	Ensemble and single models compared on AI-human task shows versatility	Performance gains marginal; feature engineering intensive
[29]	Random Forest, SVM	1500 Human texts	Word embeddings	92	Embedding-based features significantly boost classical models	Small dataset limits generalization; embedding quality dependent on pretraining data
[30]	SVM + LSTM	Mixed Genre Corpus	Word embeddings	85	Hybrid approach balances interpretability and sequence modeling	Complexity in combining models; tuning both components is challenging
[31]	Hybrid Detection Framework	GPT vs. Human news articles	Word embeddings	88	Unified pipeline demonstrates end-to-end applicability across news domains	Framework complexity may hinder real-time deployment
[33]	RoBERTa	Human vs. LLM Text Corpus	Pretrained embeddings	93	Achieves state-of-the-art results with minimal feature engineering	Large model footprint; sensitive to domain shift without further fine-tuning
[34]	BERT	Essays	Pre-Trained Embeddings	90	Strong baseline with deep contextual representations	High inference time; less efficient compared to distilled variants

Table 1. Analysis of existing studies (Results in acc %).

$$T_{clean} = \{w_i \epsilon T | \, w_i \notin (S \cup D \cup P \cup L \cup C)\} \tag{1}$$

where T is the original Text, S is a set of stop words, D contains Digits, P represents punctuation marks, L is a set of hyperlinks, C belongs to special characters, and T_{clean} is a cleaned text with all unwanted elements removed.

Lemmatization and Stemming: These terms were standardized by transforming words into their base forms, some of which are known as lemmas λ and some as root forms based on tokens t_i, defined as in Eq. (2)[42].

$$L = \{\lambda(t_i) | \, t_i \epsilon \tau(T_{clean})\} \tag{2}$$

Text normalization: This involved standardization to one case and formatting for the cases, including conversion of all to lower case, as shown in Eq. (3)[43].

$$T_{norm} = Lem(Tok(Lower(T_{clean}))) \tag{3}$$

Tokenization: To perform the analysis, the text was divided into individual work-bearing elements or tokens. This is the first step to ensure that each token recognized is subsequently used for the next step, using Eq. (4)[44].

$$T_{token} = Tok(T_{norm}) \tag{4}$$

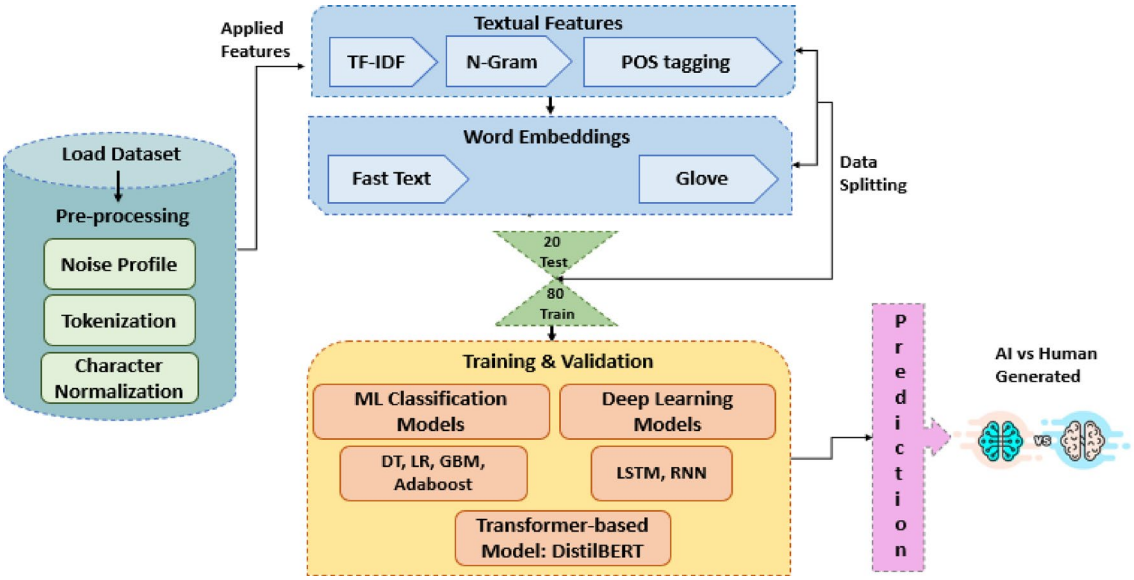

Fig. 2. Framework showing the steps of the proposed methodology.

Feature engineering

Feature engineering is likely the most important step in the NLP pipeline because it involves the conversion of the raw, textual data into forms that are more suitable and usable by machine learning (ML) or deep learning (DL) algorithms. Clustering methods can be broadly categorized into two types: numerical or categorical representations that can be extracted from textual data and deep semantic features that capture word and relation meanings[44]. The following are the methodological descriptions of the techniques employed in this research.

Term frequency-inverse document frequency (TF-IDF)

TF-IDF is a technique that is used to encapsulate the relevance of different words in a document about the completion of the document, as explicitly defined using Eq. (5). This approach yields higher values for words that are more frequent in a particular document and rarely used in total collection.

It offsets more often used do that affect words and phrases that still contain meaningful information than used for distinction. The TF component indicates the number of times a particular term is found in the document, whereas the IDF component reduces the importance of terms frequently used in the document. TF-IDF is the most useful when utilized for the bag-of-words model and the sparse matrix for all the conventional machine learning classifiers.

$$\begin{aligned} TF-IDF\left(t,d,D\right) = & \left(\frac{\left(count\left(t,d\right)+\alpha_{TF}\right)}{\sum_{w\epsilon d}\left(count\left(w,d\right)+\alpha_{TF}\right)+\gamma_{TF}.len\left(d\right)}\right)^{\beta_{TF}} \\ & * \log\left(\frac{|D|+\beta_{IDF}}{|\{d'\epsilon D : t\epsilon d'\}|+\beta_{IDF}}\right)^{\gamma_{IDF}} \\ & * \left(1+\delta_{norm}.\left(\frac{sum\left(t,d\right)}{len\left(d\right)}\right) * \left(\frac{f\left(t\right)}{\sum_{t'\epsilon d} f\left(t'\right)}\right)\right. \end{aligned} \tag{5}$$

N-grams

N-Gram is often derived by slicing text into sequences of words contiguous to each other, primarily to track the words and order of the phrases. By creating sequences of 2 or more tokens, this method can maintain the syntactic structure of words, which is usually used with weighting of features, for example, with TF-IDF, as defined in Eq. (6). Word combinations play a major role in applications such as sentiment analysis and topic modeling, where N-grams are important.

$$\begin{aligned} P\left(w_1,w_2,\ldots,w_n|\,C\right) = & \frac{\prod_{i=1}^{n}\left(TF\left(w_i,C\right)\right)}{Z} * \\ \prod_{j=1}^{n}\left(\frac{count\left(w_{j-n+1},\ldots.,w_j\right)}{count\left(w_{j-n+1},\ldots,w_{j-1}\right)+\beta_{smooth}}\right)^{\zeta_{weight}} & \end{aligned} \tag{6}$$

POS tagging
With POS tagging, we identify the register of each word in a sentence, for example, noun, verb, or adjective. This procedure identifies basic constituents possibly outlining the use of each word in view of a given sentence form, as defined in Eq. (7). For instance, further classification can be performed by determining whether a word is a subject, action or description, among others. POS offers feature creation because of the frequency of certain tags (such as adjectives for sentiment analysis). Most apparent in tasks that require assessment of the linguistic or stylistic features of a particular text.

$$P(t_1, t_2, \ldots, t_n | w_1, w_2, \ldots, w_n) = \bigcup_{i=1}^{n} P\left(t_i | t_{i-1}\right) \cdot P\left(w_i | t_i\right) \cdot \bigcup_{j=1}^{n} \sum_{t_{j-1}} \sum_{t_j} P\left(t_j | t_{j-1}\right) \cdot P\left(w_j | t_j\right) \quad (7)$$

GloVe
It incorporates global statistical data and local context, which uses the sliding window approach for training, GloVe builds a co-occurrence matrix to compute the affinity of words across the whole corpus. This leads to compact vectors where proximity preserves semantics, defined using Eq. (8), for example, king–man + woman = queen. For analogy and other tasks where word relations are important, GloVe performs exceptionally well.

$$W = \prod_{i=1}^{V} \prod_{j=1}^{V} f(X_{ij})(w_i^T \widehat{w}_j + b_i + \widehat{b}_j - \log(X_{ij}))^2 \quad (8)$$

Fast text
The embedding technique FastText was developed by Facebook AI Research to capture semantic and morphological information and is provided in a continuous vector representation within a low-dimensional space. FastText differs from traditional word embeddings in that it uses subword information, i.e., it breaks words into character-level n grams. This structure of the model enables it to deal effectively with rare words, misspellings, and morphological variations, which are particularly useful for capturing semantic shades in textual data[45]. FastText is an aggregate embedding vector constructed by summing the embeddings of a subword stack at the multilayer embedding level. The sum is then capped, yielding a robust embedding that understands the semantic similarity of two words that are not seen in the training. This powerful capability allows tasks such as text classification (e.g., detecting AI-generated vs. human-generated texts) since fast text embeddings can be used by classification models to effectively learn an exploit near subtle linguistic patterns, such as fixed lexical choices or fix morphology.

Model application

These selected models allowed us to classify the text data successfully. These include basic forms of models such as machine learning and complex forms such as ensemble learning as well as complex deep learning forms. A detailed description of the applied model is given below.

Machine learning models
Decision trees are composed of branching structures; they are trees that divide the given data into subsets according to the value of features. Inside nodes are decision circles with reference to features, and leaves depict classes on trees. The model divides the data into subsets consistently based on the exclusive feature that has the greatest information gain or the lowest Gini indices. Decision trees are easily understandable and can handle both numerical and categorical variables properly. Furthermore, logistic regression is a statistical operation used in solving binary and multiclass classification problems. The method calculates the probability of a class label using logistic transformation of linear regression of inputs. Before applying this function, the predicted values are scaled between 0 and 1, making the model useful for probability-based decisions[46]. Logistic regression assumes that there is a linear relationship between the features and the log of odds of event occurrence. It is computationally efficient and hence can be used with larger datasets more efficiently.

Boosting learning models
AdaBoost is an acronym for adaptive boosting and is an ensemble learning process in which several weak classifiers are usually combined into decision stumps to create a strong classifier. AdaBoost also operates in cycles; moreover, it assigns weights to misclassified instances, so the next classifiers revised their attention to the most difficult sample. Each time, a new weak learner is introduced, and the two outcomes are subsequently combined using a weighted sum. This approach improves the performance of the models and does not hamper the interpretability of the models to a high degree. Moreover, the gradient boosting machine is a very efficient technique and is an ensemble method that consists of decision trees arranged one after the other, with each tree acting to minimize the errors of its previous trees. In contrast, AdaBoost performs gradient descent to minimize the differentiable loss function by minimizing the residual errors. GBM's high flexibility allows it to process different loss functions for classification, regression and ranking problems. It can detect interactive and nonlinear relationships between variables[47]. Nevertheless, these methods result in GBMs that are computationally expensive and may tend to overfit; this can be remedied by optimizing the choice of hyperparameters, including the learning rate, the number of trees, and the maximum depth.

Deep learning algorithms

RNNs are particularly suitable for using sequential data because they can keep data from previous inputs in the sequence via the hidden state. In fact, each neuron of an RNN takes an input at a certain time step, and its output is then a function of the current timestep's input along with that of the previous timestep. The use of context within the information makes RNNs ideal for use in tasks that require comprehension of context, such as text classification, speech recognition, and time-series prediction. However, they have difficulties known as vanishing and exploding gradients during the back propagation process, which hinders their ability to handle long-term data sequences. To mitigate such a case, variants such as gated recurrent units (GRUs) or LSTMs are used most of the time. In another applied model, LSTMs are a refinement of traditional RNNs that are intended to improve the ability to pick up long-term dependencies due to the vanishing gradient issue. They achieve this using three gates: presynaptic, postsynaptic and postsynaptic control elements known as the input, forget, and output gates, respectively. It is through these gates that selection takes place to add, update, or prune information from the cell state. These can work with long sequences because relationships might exist among many of them, making LSTMs ideal for text data[48]. This approach is especially useful for tasks such as sentiment analysis, the translation of languages and text production[31]. Thus, this rich set of models guaranteed thorough analysis, including the use of both readily interpretable methods and complex nonlinear architectures for achieving high accuracy combined with adequate capacity for discovering complex regularities in the patterns under consideration.

Transformer-based model

The DistilBERT model, a distilled variant of the transformer-based BERT architecture, effectively detects AI-generated versus human-written text by employing self-attention mechanisms. Unlike traditional sequential models, DistilBERT captures global contextual dependencies by assigning dynamic weights to different words based on their context within a sentence or paragraph, as shown in Fig. 3. This allows the model to recognize subtle linguistic nuances, semantic coherence, and stylistic patterns specific to either human or AI-generated text. During training, DistilBERT fine-tunes its pre-learned embeddings on labeled data, learning subtle discriminative features such as syntactic structures, repetitive or predictable patterns, and coherence variations[49]. The model's attention layers focus dynamically on contextually relevant tokens, allowing it to discern nuanced textual differences effectively, resulting in exceptionally high identifying accuracy.

Dataset

The dataset for this research study was taken from a famous and widely used online source, Kaggle. The dataset consists of 500,000 essays written by both humans and AI-based large language models (LLMs). Therefore, this helps us to have one target class, and having two values of human vs AI makes this a binary classification research problem. The content was only in the English language. The dataset is the latest and was prepared in 2024. The dataset is large, newly updated and properly labeled, which makes it a suitable and standard dataset for carrying out empirical analysis in this research domain.

Experiment details

For the experiments were carried out within a Python environment, supported by text-processing, embedding production, machine learning, and deep learning libraries on machines designed for performance and speed,

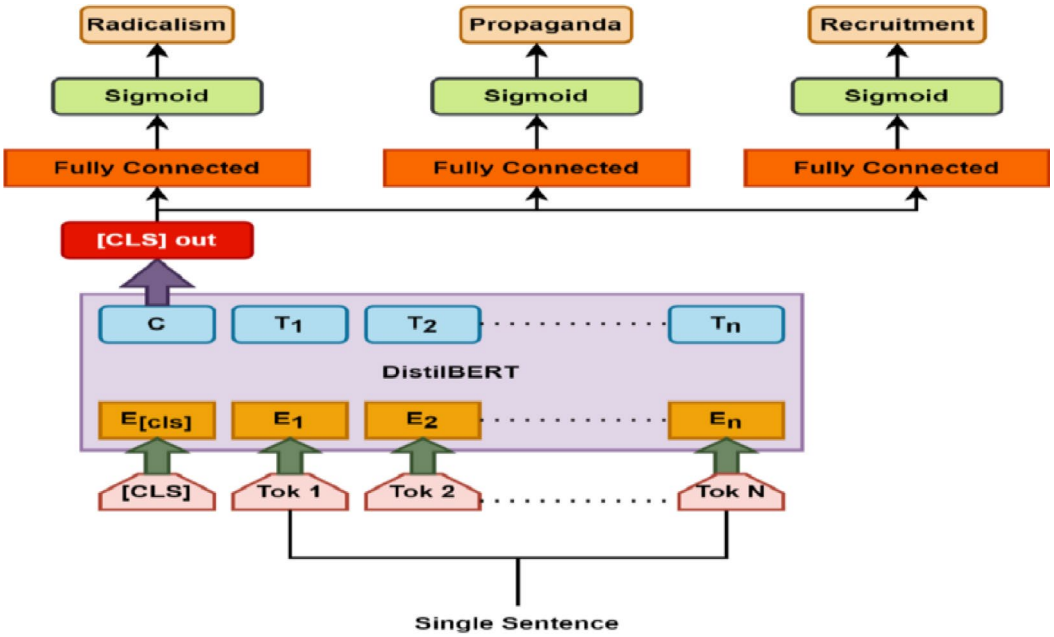

Fig. 3. Working of the DistilBERT Model in NLP Tasks.

as components displayed in Table 2. Using NLTK, spaCy, scikit-learn, Genism's libraries for embedding such as FastText tools, as well as glove-python-binary, the text was preprocessed, and features were generated. The DistilBERT fine-tuning task was done in PyTorch and Hugging Face's Transformers library. Both statistical analysis and visualization in the project were carried out with SciPy, statsmodels, Matplotlib, and Seaborn. The processed and tested the models on Windows environment Tesla V100 GPU, a 24-core Intel Xeon CPU, and substantial memory.

Performance metrics

To avoid bias in selecting these models, the standard performance of the models was assessed by using popular parameters. The above metrics provide information about different facets of the models' prediction ability and thus provide a full evaluation.

Accuracy

The accuracy calculates the ratio of the instances that were predicted correctly against the total number of instances that were predicted. This approach has a generic application when providing an approximate estimate of the performance of the model in question in terms of its correctness in classification. However, accuracy can be compromised by poor performance on minor classes, particularly in imbalanced dataset scenarios[50]. The accuracy was calculated using Eq. (9):

$$Accuracy = \frac{(TP + TN)}{TN + FN + FP + TP} \tag{9}$$

where TP, TN, FP and TP stand for true positives, true negatives, false positives and false negatives, respectively.

Precision

Precision limits efforts toward predicting positives by including only the correctly identified positive instances as true positives, which are divided by the total number of positive instances identified. This approach is especially helpful when false positive results mean high costs, and people are dealing with issues such as fraud or disease. Thus, the high precision of the model guarantees that all the predicted positive cases are accurate and relevant[51]. The precision can be computed using Eq. (10):

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Recall (sensitivity)

Recall focuses on the power of the model with respect to the capacity for correct recognition of all related cases from the set. It is expressed as the ratio of true positive rates to true positive plus false negative rates. High recall is important in cases where failure to capture a positive example is a disaster, e.g., checking people for a disease or surveillance[52]. The recall can be calculated using Eq. (11):

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

F1-score

The F1-score is an accuracy metric relating to the precision to the recall of the F1-score formula and can alleviate the problem of simply selecting either precision or recall. This approach is particularly helpful in cases of clearly imbalanced classes or when decisions over the level of necessary precision/recall are made. A high F1-score indicates that the model is adequate both for indicating relevant instances and for maintaining the accuracy of the predictions[53]. The F1-score, also known as the F-measure, can be calculated using Eq. (12):

Component	Specification
Programming language	Python 3.9
Deep learning framework	PyTorch 1.12.1
Transformer library	Hugging Face Transformers 4.21.0
Embedding libraries	genism 4.2.0 (FastText), glove-python-binary 1.2.0
Text processing	NLTK 3.7 (tokenization, POS tagging), spaCy 3.4.3
Machine learning	scikit-learn 1.1.2 (TF-IDF, LogisticRegression, DecisionTree, AdaBoost, GBM)
Statistical testing	SciPy 1.8.1 (t-tests, ANOVA), statsmodels 0.13.2 (chi-square tests, McNemar's test)
Visualization	Matplotlib 3.5.2, Seaborn 0.11.2
Hardware (GPU)	NVIDIA Tesla V100 (16 GB HBM2), CUDA 11.6, cuDNN 8.3
Hardware (CPU)	Intel Xeon Gold 6248R (24 cores @ 3.0 GHz)
System memory	256 GB DDR4 RAM
Operating system	Windows

Table 2. Detailed description measures of machine requirements for experimentation.

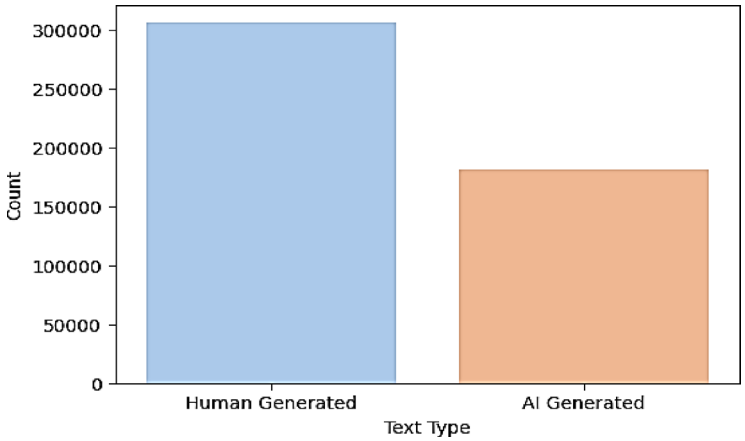

Fig. 4. Dataset Class Distribution.

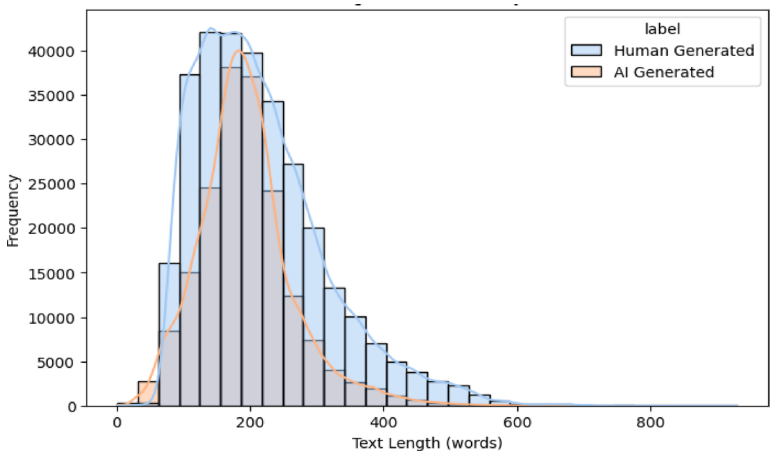

Fig. 5. Text length distribution by class.

$$F1 - Score = \frac{2(Pre * Recall)}{Pre + Recall} \tag{12}$$

By applying this multiple criterion model assessment approach, the strengths and weaknesses of the models were revealed, allowing for a comprehensive, hence, bias-free assessment with respect to the given classification tasks.

Results and discussion

This section examines the results of the proposed methodology based on ML and DL algorithms with feature engineering techniques to distinguish the content. These findings highlight the capabilities and challenges of models based on textual patterns in classifying text. These results not only show the model robustness and adaptability in handling complex text data but also provide a strong baseline for future research in detecting or classifying textual data in real-world applications.

Exploratory data analysis

Figure 4, which shows the class distribution bar chart, shows that we have an imbalanced dataset in which human-generated texts (over 300k) are more common than AI-generated texts (approx. 180k). This class imbalance makes it possible for the training of robust classifiers to be biased toward the majority class (human-generated content); thus, employing strategies such as class weight balancing or resampling to avoid training biased models is important.

The text length distribution histogram shown in Fig. 5 clearly illustrates how similar and different AI-generated and human-generated texts are about length. Typically, textual samples span lengths of moderate length and are also centered between each distribution at approximately 200 words, suggesting that these distributions are mostly about there, whatever they are gathering about, and they are mostly centered around that; however, they are also taking up about word samples. Nevertheless, there are still slight differences, as the distributions of AI-generated text are narrower with a more rapid decay in frequency toward longer lengths,

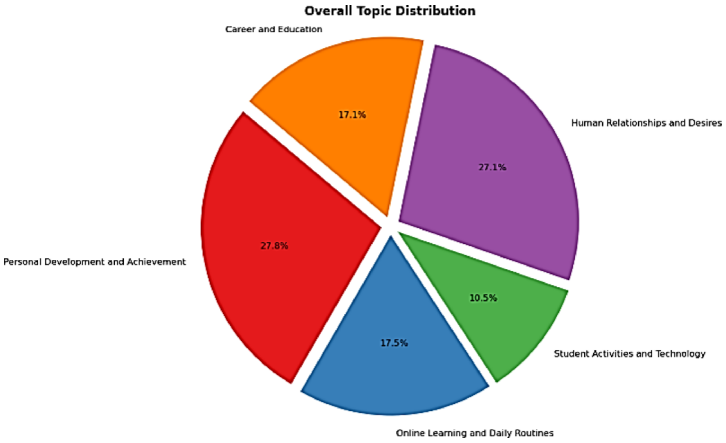

Fig. 6. Topic Modeling Distribution.

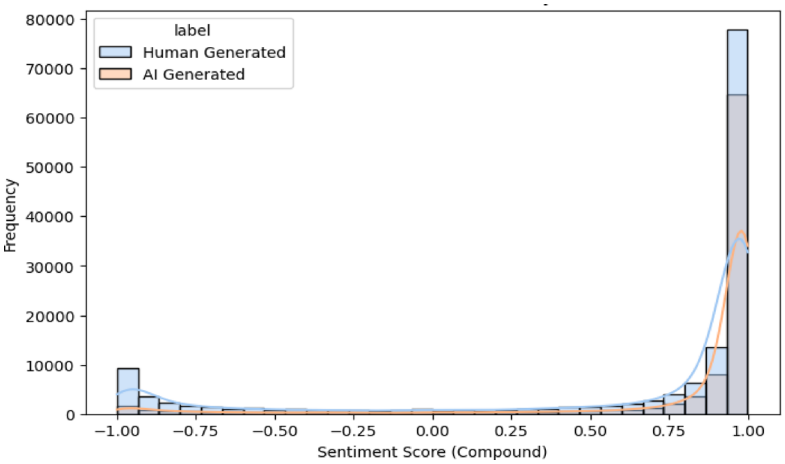

Fig. 7. Sentiment Score Distribution by Class.

suggesting that the AI-generated text is more consistent at longer lengths. Such length-based insights could help future models evolve to include length in their discriminative feature for classification.

A graphical representation of the topic distribution pie chart in Fig. 6 shows the diversity of the textual topics, which include categories such as human relationships (27.1%), personal development (27.8%), and career and education (17.1%). The fact that the trained model is exposed to different thematic areas means that classifying text authenticity in several domains does not limit its ability to generalize and classify text authenticity with excellence.

Figure 7 shows the sentiment scores (compound sentiment) of AI-generated and human-generated textual contents; this is the first aspect of identifying text characteristics. Sentiment analysis is a form of measurement of emotional tone as a value that ranges from negative (-1) to neutral (0) to positive (+1). We find that texts generated by both AI-generated and human-generated texts have sentiment scores indicating extremely positive sentiment scores, with those skewed significantly toward the positive end, indicating that most of the text, both from the AI-generated and human-generated classes, represents affirmative or positive sentiment. However, such differences exist—human written content has more spread sentiment, including a relatively higher fraction of negative sentiment, and AI-generated texts have a much tighter range around positive values. Possible reasons for such sentiment homogeneity in AI-generated text include predictable emotional and stylistic patterns, which provide subtle yet important features for advanced identification models to exploit. However, human-generated texts show greater diversity and range, ranging from more nuanced and diverse to sometimes very negative sentiments. These differences imply that transformer attention mechanisms can be improved with sentiment features and that sentiment features are meaningful inputs for improving model accuracy. This is partly because sentiment was clearly separated, and sentiment distributions provide additional valuable complementary features that help the transformer model's strong predictive capability that has been previously observed.

The word cloud visualizations themselves provide further insight in Fig. 8, both at the overall level and by themselves for the AI-generated versus human-generated texts. The prominent terms in all the samples included "elector college," "cell phone," "driverless car," "extracurricular activities," "popular vote," and so forth, indicating

shared thematic content. Interestingly, AI and generated texts have a large overlap of vocabulary in word clouds, making learning a simpler identifying model difficult. Nonetheless, there are likely unique frequency and structural differences that enable the proposed transformer-based model, which has a more advanced attention mechanism, to distinguish a given pattern well. Finally, these exploratory analyses reveal dataset complexity, class imbalance, thematic variety etc., to underline the importance of a balanced sampling strategy and proper feature engineering. This implies that there is no right to suggest the use of sophisticated transformer models that can capture very subtle semantic and contextual patterns that are needed for achieving high-accuracy AI-generated text identification.

Results with textual features

In detail, the results of various shallow machine learning (decision tree and logistic regression) and ensemble (AdaBoost and GBM) learning models trained on different textual (TF-IDF, POS tagging, and N-Gram) representations are compared to differentiate between human-generated and AI-generated textual content. Among the shallow machine learning models, logistic regression outperforms decision tree regression in terms of accuracy across feature types. As displayed in Table 3, for the TF-IDF features, logistic regression had 88% accuracy, whereas decision tree had 79%, which shows that the likelihood that logistic regression can tolerate linear relationships between textual features and labels. This trend implies the robustness and appropriateness of the logistic regression model for textual data, especially when the features are sparse and high dimensional (as is usually the case with TF-IDF vectorization). However, the decision tree performed markedly better with N-Gram features (with an accuracy of approximately 89%) than with TF-IDF and POS features (both approximately 77% to 79%), in which the decision tree was much more sensitive to textual sequential patterns than to computational weighting or syntactic structure.

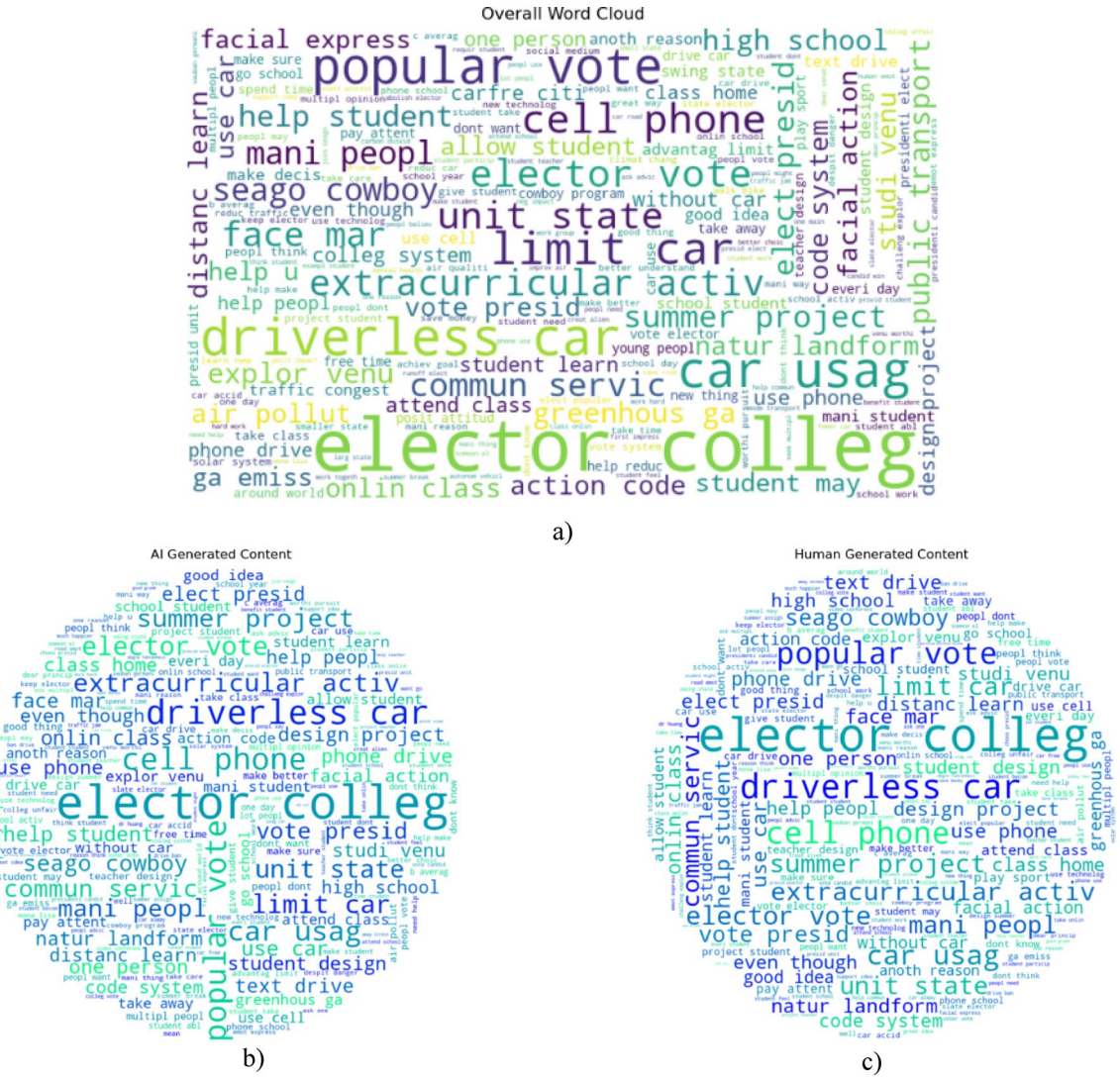

Fig. 8. Most Frequent words in dataset (**a**) overall wordcloud (**b**) showing AI-Generated Content (**c**) Human Generated Content.

Models	TF-IDF					POS					N-Gram				
	Acc	Pre	Re	F1	AUC-ROC	Acc	Pre	Re	F1	AUC-ROC	Acc	Pre	Re	F1	AUC-ROC
Shallow machine learning models															
DT	0.79	0.79	0.79	0.79	0.79	0.77	0.77	0.77	0.77	0.69	0.89	0.90	0.89	0.89	0.77
LR	0.88	0.85	0.88	0.88	0.85	0.86	0.88	0.86	0.86	0.82	0.82	0.82	0.82	0.82	0.80
Ensemble learning models															
ADB	0.86	0.85	0.84	0.85	0.91	0.85	0.85	0.85	0.85	0.80	0.85	0.82	0.83	0.85	0.79
GBM	**0.91**	**0.93**	**0.90**	**0.91**	**0.88**	**0.92**	**0.90**	**0.89**	**0.91**	**0.92**	**0.89**	**0.87**	**0.91**	**0.91**	**0.88**

Table 3. Summary of the results obtained using the ML models. Significant values are in bold.

GloVe	GloVe					Fast text				
GloVe	Acc	Pre	Re	F1	AUC-ROC	Acc	Pre	Re	F1	AUC-ROC
RNN	0.82	0.81	0.83	0.82	0.76	0.88	0.85	0.82	0.88	0.80
LSTM	**0.93**	**0.94**	**0.90**	**0.92**	0.88	**0.89**	**0.91**	**0.90**	**0.91**	**0.85**

Table 4. Analysis of results using DL models. Significant values are in bold.

Furthermore, the ensemble models, which aggregate over a single weak learner as in the simplest forms of classification, outperformed the ensemble learning models in terms of detecting textual authenticity. GBM consistently showed the strongest results overall, even more remarkable given its results with TF-IDF features of 91% accuracy, 93% precision, 90% recall and 91% F1 score. These metrics clearly suggest that GBM exploits the power of TF-IDF since using statistics ensuring the importance of unusual and discriminative terms is beneficial in detecting subtle textual patterns in favor of being written by either a person or bot. Additionally, POS (92% accuracy) and N-Gram (89% accuracy) were retained for use in GBMs, which also exhibited good performance; however, these two methods exhibit clear versatility and robustness because of their ability to adapt to different textual representations, syntax-based (POS) or sequence-based (N-Gram) representations. AdaBoost demonstrated relative consistency in terms of the results, with feature sets (approximately 85% accuracy, precision, recall, and F1-score) that stabilized and were relatively insensitive to changes in features. However, such high performance of GBM was not replicated by AdaBoost, which may be due to its limitations on noisy or less discriminative features set factoring AdaBoost's limitations from its iterative boosting strategy to discover useful insights based on diverse textual representations. TF-IDF was found to perform the most critically in terms of effective feature types, especially in extracting unique terms that coherently classify textual origins (AI vs human). Our analyses revealed that N-Gram features, as well as other sequential linguistic pattern-capturing features, performed well regardless of the language variation and for capturing sequential linguistic patterns, while POS tagging performance was relatively weaker yet stable and thus more supplementary information than was available from individual discriminators.

Overall, these findings show that the ensemble models are superior in terms of their predictive capacity, especially for GBMs with TF_IDF features, for determining whether a text is generated by AI. However, the extensive evaluation verified this claim and supported that ensemble approaches, especially gradient boosting-based approaches, prefer textual features better than shallow models. Therefore, these results provide strong evidence for the use of such models to authenticate textual sources in practice, highlighting important aspects that should be taken into consideration in other studies on the exploitation of more nuanced representations of linguistic and statistical features to increase detection capabilities.

Results with deep features

Table 4 presents the performance metrics—accuracy, precision, recall, and F1-score—of two deep learning methods (LSTM and RNN) using two different embedding methods—FastText and GloVe—for classifying texts based on whether they were generated by humans or not.

The results clearly demonstrated that LSTM outperformed RNN on both embedding techniques because these models have strong sequential memory and context retention ability, which are particularly helpful for discriminating against AI-generated and human-generated text. LSTM with GloVe embeddings specifically performed very well and provided the highest accuracy (93%), precision (94%), recall (90%), and F1 score (92%). The strong performance achieved by GloVe embeddings in capturing global semantic information highlights their ability to encode this information in such a way that allows the LSTM model to distinguish the subtle textual HNPs that characterize human versus AI-generated content. For instance, LSTM still performed robustly when added to FastText embeddings (accuracy of 89%, precision of 91%, recall of 90%, F1-score of 91%) but was less effective than when GloVe embeddings were used. Although morphological variations, which are common in human writing and may include the repetitive nature of AI-generated content, can be captured by FastText embeddings that extend subword-level information, they also provide strong predictive abilities, despite the embeddings being designed for recognizing such patterns. When the RNN model was paired with GloVe embeddings, it exhibited reasonable predictive power but was also lower than that of the other models, with an accuracy of 82%. Nevertheless, the learning performance greatly improved when using a FastText embedding

accuracy of 88%, which means that character-level or subword information embedded in FastText somehow performs better than treating such information as raw sequences that do not utilize their context, especially when just using global semantic embeddings (GloVe) in simpler recurrent neural architectures. One important inference from this comparative analysis stems from the superiority of the LSTM model in terms of prediction; these models are inherently capable of handling longer textual dependencies and complex linguistic patterns through their gated mechanisms and memory cells. This further supports the fact that semantic contextual information is very important in terms of performance advantage, and it helps discriminate subtle variations in text authorship in GloVe embedding-based models with LSTM architectures. On the other hand, FastText, although with slightly lower performance, provides additional support for the idea that morphological features and subword information are crucial to text classification tasks.

These findings strongly indicate that LSTM-based deep learning models with semantic embeddings (GloVe) are very strong and appropriate for identifying AI-generated textual content versus human written content. This thorough evaluation confirms how deep learning models can exploit complex language patterns and suggests its great potential to reach high accuracy in textual authenticity assessment in real situations.

Results with the transformer-based model

The results also showed that the transformer-based model DistilBERT performed well in correctly differentiating between AI-generated text and human-generated text. Notably, the DistilBERT model achieved an extremely high accuracy of 0.98, a precision of 0.92, a recall of 0.96, and an excellent F1 score of 0.98, as shown in Fig. 9. These are extremely impressive results that indicate the model's ability to describe in detail and meaning that are needed for the classification process.

Self-attention mechanisms enable DistilBERT to coercively weigh the relevance of words and their contextual relationships in texts. The architecture of this attention-based architecture is such that we can effectively recognize or understand subtle patterns, linguistic subtleties and contextual cues that otherwise largely distinguish AI-generated content from human-generated material. The high recall (0.96) here is meaningful because the model successfully identifies almost all true cases as AI-generated content, and it is both sensitive and reliable as a detection mechanism. Moreover, the precision score (0.92) indicates that the tool retains good discriminatory power, recognizes human content very rarely as AI is generated, and is crucial for applications that need very high trust and credibility, as supported by using confusion matrix in Fig. 10.

Figure 11 shows the training and validation accuracy of the proposed classification model across four segments (from 1 to 25 epochs, 26 to 50 epochs, 51 to 75 epochs, and 76 to 100) for predicting whether a given piece of textual content has been generated by machines (AIs) or humans. The accuracies are shown in the form of gradient-colored bubbles per subplot, where each bubble accurately reflects the trend unequivocally and vividly. The training accuracy initially (epochs 1–25) is initially sharp and increasing but then increases by approximately one hundred percent from approximately 86% (epochs 26–30). Moreover, the validation accuracy follows an inverse path tracing downward and bottoming out around the midpoint, implying early stages of overfitting, generalization problems, or a lack of the model at adequately learning and apprehending significant patterns from the data point of view. In epochs 26 to 50, there is an interesting occurrence: not only does the training accuracy peak, but it also starts sharply declining while the validation accuracy grows steadily to 85%. A better generalization of the model is shown by this crossing of training and validation accuracy curves. A decrease in training accuracy might be a signal to change the architecture of the model or learning rate so that we do not overfit the model and that the model trains less specific textual features.

At approximately 51–75 epochs, another similar and opposite trend occurs, where the training accuracy starts to increase once more and remains at approximately 82%, while the validation accuracy decreases from its peak to approximately 74%. This means that there is a possibility of overfitting recurrence where the model starts to suit the peculiarities of the training set too well and cannot generally extend unseen textual patterns

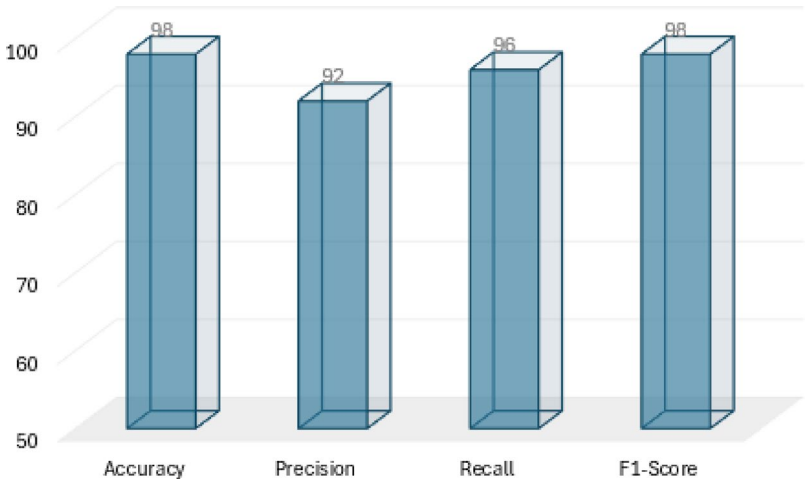

Fig. 9. DistilBERT Performance Analysis.

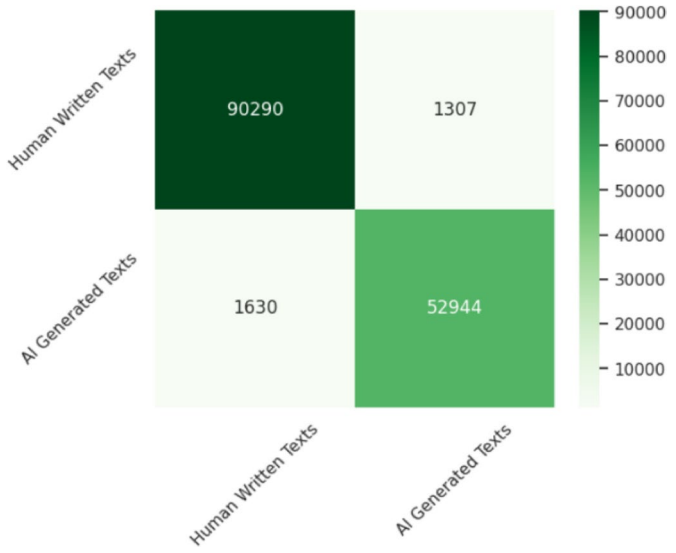

Fig. 10. Confusion matrix of proposed model.

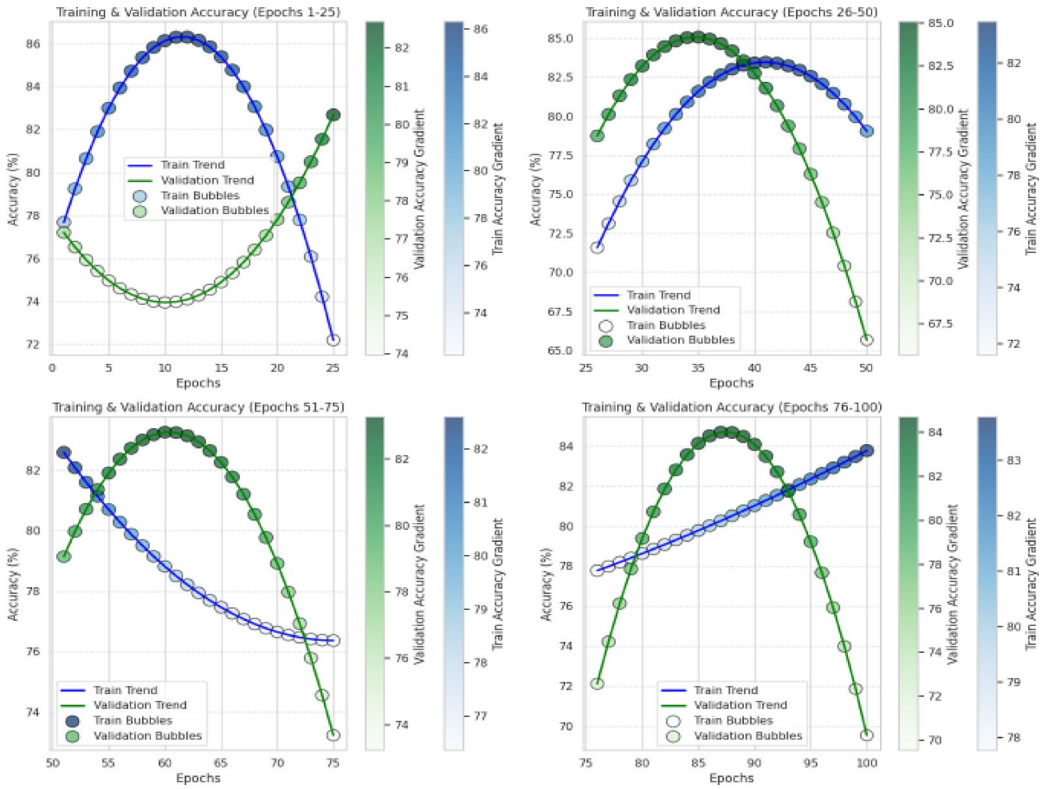

Fig. 11. Accuracy analysis of the model during the draining-over-epoch stage.

effectively. Such fluctuations are a model instability indicator and might imply a need to hyperparameter tune or apply regularizing techniques, displayed in Table 5. As we see in the final epochs (76–100), the model converges to stable behavior where the training accuracy stagnates at approximately 84%. The validation accuracy peaks and then decreases dramatically at the same time. It is evident from the repeated pattern of improvement in training and subsequent decline in validation that the model learned something—however limited—but failed to generalize whenever the validation data changed. Although the model adapts well to training data features, its ability to be sensitive and overfitted needs to be more regularized, and perhaps early stopping must be deployed.

Overall, this figure gives a very good picture of the complexity of developing stable, generalized performance in deep learning over extensive training periods. While the model shows high learning capacity on the training

Hyperparameter	Value	Description
Learning rate	2e-5	Small rate to fine-tune transformer layers without catastrophic forgetting
Batch size	16	Balanced between GPU memory footprint and stable gradient estimates
Number of epochs	4	Sufficient for convergence given early-stopping based on validation loss
Max sequence length	256	Truncates/pads inputs to limit compute and memory per example
Weight decay	0.01	Regularization to prevent overfitting
Dropout rate	0.1	Applied in attention and feed-forward layers to improve generalization
Gradient accumulation steps	2	Simulates larger batch sizes without increasing memory usage
Class weights	{0:1, 1:1.8}	Adjusts loss to emphasize minority (AI-generated) class, counteracting the 80:20 imbalance
Warmup steps	500	Gradually ramps up learning rate at start of training for stable optimization
Maximum grad norm	1.0	Clips gradients to prevent exploding gradients

Table 5. Hyperparameter settings of proposed model.

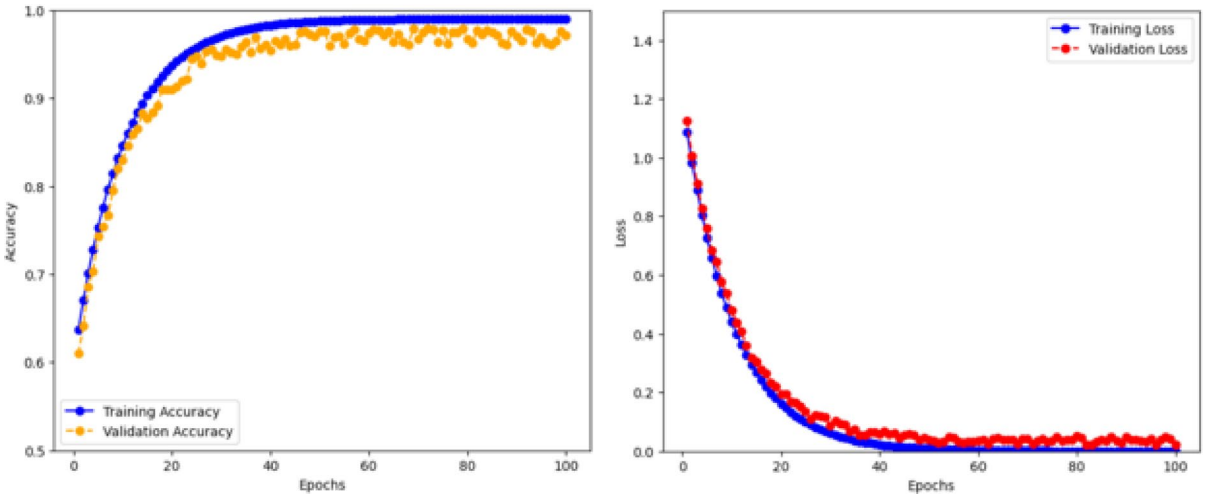

Fig. 12. Model accuracy and loss analysis.

data (as per its periodic peaks in accuracy), there are repeated divergences between the validation and training accuracy, signifying that it has a difficult time generalizing learned features to unseen textual texts. For these concerns, the hyperparameter tuning plays a pivotal role for addressing such concern of overfitting, class imbalance, activation function etc., the distilled transformer model consisted of several key hyperparameter settings which needed careful manual tuning to balance the tradeoff between convergence speed, generalization, and efficiency. The use of a low learning rate and slow warmup scheme allowed the pretrained weights to be optimized without catastrophic forgetting. Values of batch size and gradient accumulation were determined to fully utilize GPU resource and avoid memory overflow, and sequence length was truncated to concentrate on the most informative text segments. Regularization methods—weight decay and dropout—were used to prevent overfitting with gradient-norm clipping to aid convergence. Crucially, used class-weighted loss to balance the skewed human-to-AI sample ratio to enable the rare-class examples to contribute proportionally to the gradient signals, hence avoiding the domination of the majority class. Early-stopping criteria driven by validation performance further reduced the extra training time after accuracy saturation and enabled a robust and fast detecting model to be obtained.

To increase stability and predictive reliability, strategies such as early stopping, learning rate scheduling, dropout, or additional regularization mechanisms could be used, as shown by the overall loss and accuracy analysis in Fig. 12. The accuracy trends depicted make it clear that continuous performance monitoring during training is essential for maximizing the identification of textual content generated by an AI.

Thus, the model is evaluated for unseen data using the so-called validation set, which is an independent subset of data not seen by the training process, to measure how well the model can generalize and is robust to training. The validation accuracy is the proportion of how correctly the samples from that set are classified; this is information on how good the model is at capturing the patterns without overfitting the training data. This plot, as shown in Fig. 13, provides a visual representation of the validation accuracy of our model in distinguishing AI-generated textual content from human-generated content. The validation accuracy usually fluctuates widely, although it can alternate between values flowering at approximately 60% and those fluctuating at approximately 100%. Given that the trend line is smoothed, cyclic variability can be emphasized with a line, and it can be observed that there are periods over time where generalization improved followed by a decline, indicating

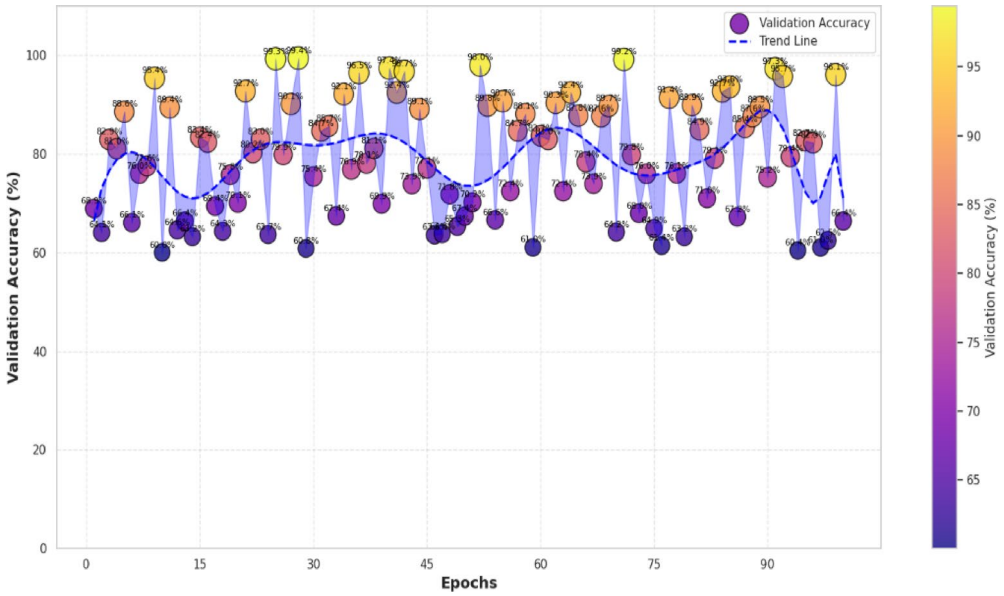

Fig. 13. Validation accuracy analysis of the model.

unstable generalization behavior. Repeated fluctuations act as sources of ambiguity in the learning rate or complexity of the dataset or feature extraction variability across epochs. The model attains high peaks adjacent to 95–100%, indicating great learning capacity; however, numerous dips below 70% suggest some major difficulties in regularly utilizing found highlights to reconfirm data. Overall, however, the performance of the model is mildly good overall but highly inconsistent, probably indicating that additional tuning to the regularization or early stopping or possibly changing the model architecture is needed for stabilized and more reliable predictions.

Additionally, for a deeper analysis of model predictive power, Fig. 14a,b shows the model performance and prediction ability of the trained model when distinguishing between the textual content from AI and human data generation, specifically for estimating confidence and accuracy on models and samples of text. Figure 14a shows the model's confidence in predictions over several epochs, wherein it fluctuates but generally has high confidence. For the most part, the model surmounts confidence of more than 90% in both AI-generated and human-written texts. However, there are noticeable drops in confidence, sometimes approximately 60%, which appear to be an indication of variability and varying degrees of uncertainty. This fluctuation may be attributed to the complexity or ambiguity of some of the textual samples, illustrating that while the model does well to identify many texts, some subtle aspects continue to be difficult. Figure 14b clearly shows a different outcome in terms of the predictions on individual textual samples; green bubbles represent correct predictions. In addition, in this case, it is impressively high, with an average prevalence greater than 89% and often greater than 97%. The model shows good performance in terms of separating AI-generated texts from human-generated texts, which further asserts the ability of the model to generalize to unseen data samples. The fact that there are no incorrect predictions on these specific examples indicates that the model is reliable and accurate for classifying these examples.

In addition to training time, the curve (blue line) and validation accuracy curve (orange line) of DistilBERT provide useful insights into its learning and computational efficiency, as shown in Fig. 15. In just the first 10 epochs, the training accuracy went from below 50% to above 95%, indicating that the model quickly learns the dominant textual patterns distinguishing between AI and human text authorship. The validation accuracy followed suit, going above 80% by epoch 10 and stabilizing between 90 and 94% after that; this is a large range of generalization with little amount of overfitting, despite the number of epochs the model is trained on. The plateauing of both curves after epoch 20 demonstrates that much of the useful representation learning occurs early on in Fine-tuning; that is, more epochs after that begin diminishing returns in model improvement and imply that an early-stopping criterion could be used to save computing time without any impact on accuracy. The dominant computational time grows approximately by $O\left(E \times N \times L^2\right)$ in terms of the number of epochs E, the total number of tokens N, and the max length L, due to self-attention's quadratic dependency on the sequence length. In practice, it trained for 100 epochs in roughly 2 h (≈ 72 s/epoch), but validation accuracy saturates around epoch 20, indicating possible reduction in training time by up to 80% with early stopping. Furthermore, the memory consumption grows linear with respect to the batch size (B) and the sequence length (L); as each token's embedding and D-dimensional intermediate activations need to be stored in GPU RAM; in reality, the batch size of 16 with a sequence length of 256 consumed around 5.8GB (≈ 19.2% of a 30GB GPU), indicating the model's cost effectiveness for large-scale text processing. This shows that from a time-complexity perspective, the run time for each epoch scales basically linearly with the number of training samples, sequence length, and the number of layers in the transformer; however, the model finishes with modest GPU benchmarks of 5.8 GB (19.2% of total available memory), demonstrating DistilBERT's efficiency as a compressed transformer.

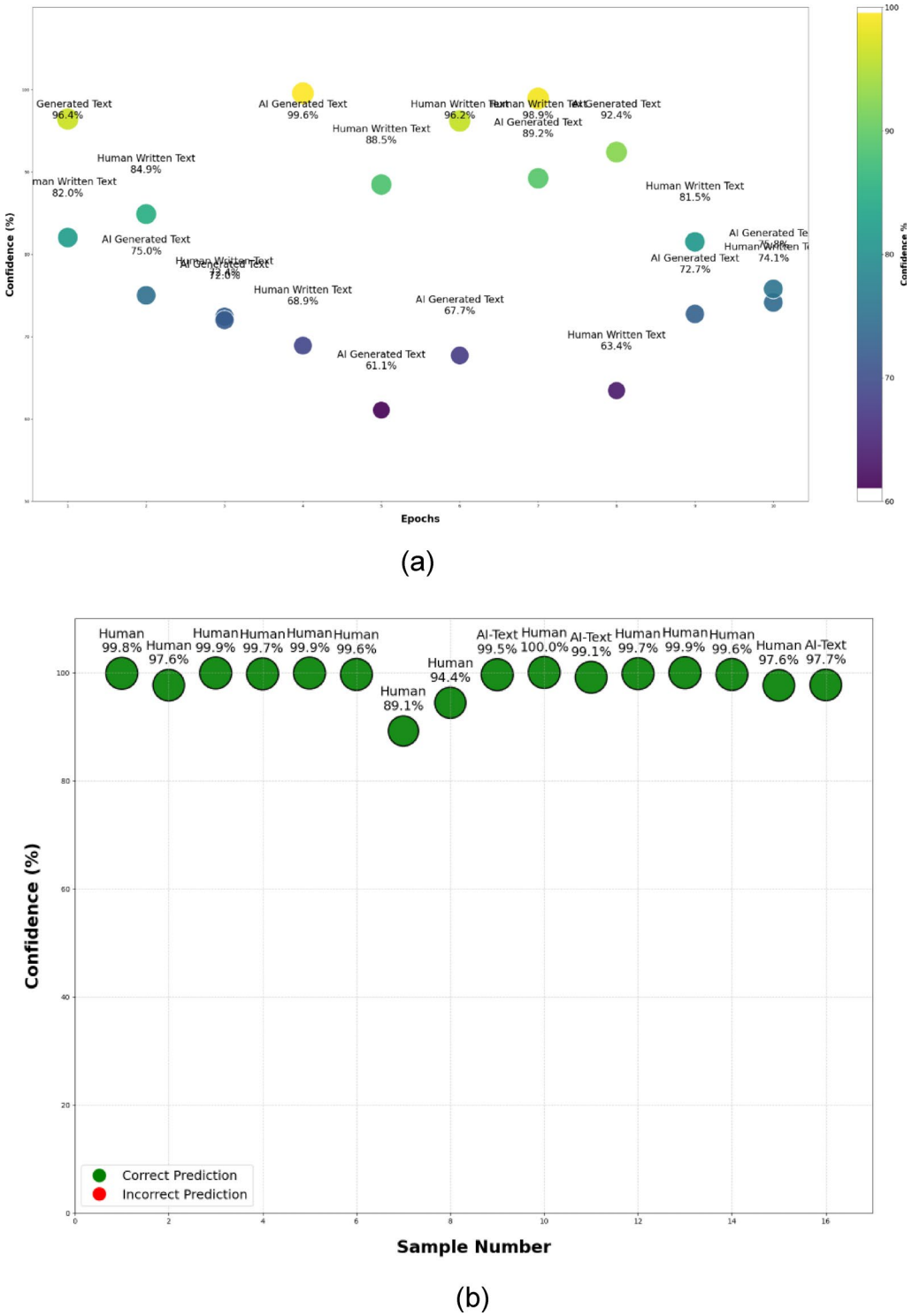

Fig. 14. (**a**) Confidence analysis based on prediction with label test data. (**b**) Confidence analysis based on prediction with random samples.

The low resource requirements allowed for rapid experimentation: completing 100 epochs only took a few hours, meaning you could develop an idea on DistilBERT and test it on both its base model and additional epochs Tesla V100. In conclusion, not only does DistilBERT provide highly formal detection performance at fast speeds and attenuated hardware complexity.

Finally, Fig. 16 contains textual samples along with their respective actual and predicted labels and confidence scores for further qualitative understanding. The predicted and actual origin (AI vs. human) is clear, coherent and highly put very high confidence (ranging mostly from 89–99%). The model can distinguish between

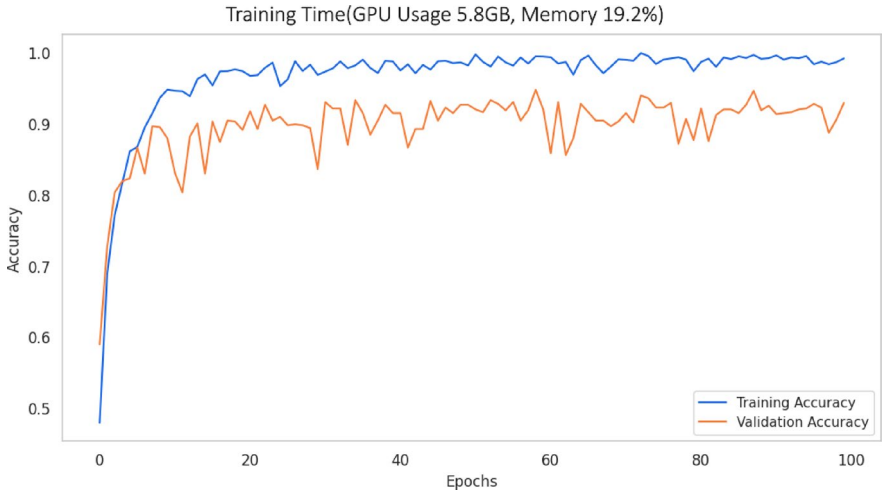

Fig. 15. Analysis on training progress consumption.

Fig. 16. Classification results of human-written and AI-generated text with confidence scores.

peculiarities such as syntax, coherence, topic consistency, and structure common to human-generated texts and peculiarities such as recurrent patterns, redundancy, or structural anomalies observable in AI-generated texts.

Collectively, these results strongly affirm the model's effectiveness and reliability in detecting textual authenticity. Both high confidence levels over most predictions and high accuracy in correctly classifying challenging examples indicate that the model captures its semantics and context well, which shows good promise for practical deployment applications that require text authentication, including academic integrity verification, content moderator and misinformation. Although confidence varies between epochs on an occasional basis, there is still room for improvement to ensure that confidence scores remain high for all types of textual content.

Additionally, statistical tests also applied to determine which attributes in the text distinguish AI-generated content from human-written content. Figure 17 shows that, based on statistical analysis, every feature tested is significantly different between AI and human texts, but nothing is revealed about their relative importance or if some features are not very important. Using two-sample T-tests, the mean values of text length, word count, average word length, unique-word ratio, stop-word count, and punctuation count were examined to check if they were significantly different in the two classes. Using Chi-square tests of independence to check if the occurrence of formal connectors words. Additionally, a one-way ANOVA analysis on word counts showed differences in

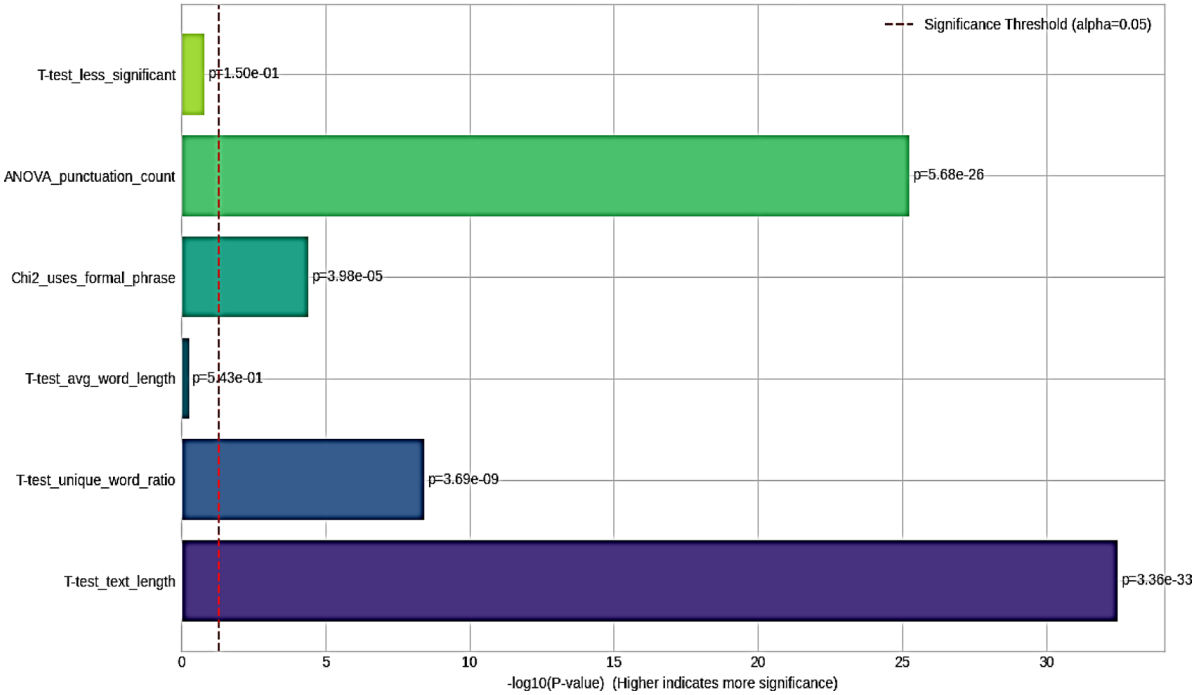

Fig. 17. Comparison of $-\log_{10}$(p-values) for multiple statistical tests on training-set features, illustrating that all evaluated text attributes significantly differ ($p < 0.05$) between AI-generated and human-written texts.

overall text between classes. The main factor influencing the model is the average document length, as it has the largest $-\log_{10}(p)$ value of around 32 ($p \approx 3.36 \times 10^{-33}$). This is consistent with our observations from DistilBERT, which often gives more attention to sentence and paragraph breaks when making decisions. Moreover, the analysis of punctuation shows a $-\log_{10}(p)$ of ~ 25 ($p \approx 5.68 \times 10^{-26}$), proving that the model pays close attention to punctuation, which is a key feature. This test also shows that the difference between AI and human text is highly significant ($-\log_{10}(p) \approx 8$, $p \approx 3.69 \times 10^{-9}$), and it is due to AI text often having less variety in its vocabulary. By transforming the p-values into $-\log_{10}$ and checking them against a standard level of $\alpha = 0.05$, managed to pinpoint which features are truly different and therefore useful for the detection process. The Fig. 17 shows the $-\log_{10}$(p-value) for six statistical tests and marks the $\alpha = 0.05$ threshold with a red dashed line at $-\log_{10}(0.05) \approx 1.3$. The bar is above line, that means the difference in writing between AI and people is significant at the 5% level or higher—and bars much longer than this mean the difference is even stronger. When a difference is below the line, this means cannot reject that there is no difference. The text length proves to have the highest significance ($p \approx 3.36 \times 10^{-33}$, $-\log_{10} \approx 32$), showing that AI texts tend to be either longer or shorter than human texts. The next unique-word-ratio T-test ($p \approx 3.69 \times 10^{-9}$, $-\log_{10} \approx 8$) proves that there is a strong distinction in word choices. Texts written by humans include a bigger collection of words in relation to their length. The evaluation of punctuation shows significant differences in their use by the two individuals. The results of the Chi-square test for formal phrases ($p \approx 3.98 \times 10^{-5}$, $-\log_{10} \approx 4.4$) show they are present to a lesser extent, suggesting that AI writers may include formal connectors more often than humans do. Conversely, the average-word-length T-test ($p \approx 0.543$, $-\log_{10} \approx 0.27$) and the deliberately labeled "less significant" T-test ($p \approx 0.15$, $-\log_{10} \approx 0.82$) both fall below the red line, meaning that neither average word length nor this feature successfully separates AI-written text from human texts.

The reason DistilBERT pays little attention to these attributes is because they are not significant. The Chi-square test for formal phrase usage is significant but less so than the other two factors. The model uses a moderate amount of conjunction tokens like "however" in its attention maps, as shown in the result. All these statistical analyses agree with the observed accuracy, proving what matters most to the DistilBERT classifier and what can be ignored. With these results, it becomes easy to decide which features are truly useful and which can be set aside.

Furthermore, the statistical significance chart and its results show why DistilBERT is so effective in classifying the test data. Based on the unseen data, the statistical test applied to text length, punctuation count, unique-word ratio, and the use of "however" are highly discriminative, as they are well above the significance threshold, as shown in Fig. 18. The results are useful since they show both which attributes contribute most to the model and the relative strength of these attributes. Based on the T-test ($T \approx 84.8$, $p \approx 0$) and ANOVA ($F \approx 21{,}131.6$, $p \approx 0$), the size of AI-generated text is much different from human-written text. AI sentences are usually of the same length, while human authors often produce sentences of different lengths. Because DistilBERT's attention heads are aware of document boundaries and token counts, its learned representations are better separated. It becomes evident from the T-tests that there is a significant difference between human and AI-generated text in terms of stop-word use and unique-word ratio. DistilBERT learns to focus on different patterns of word co-occurrence

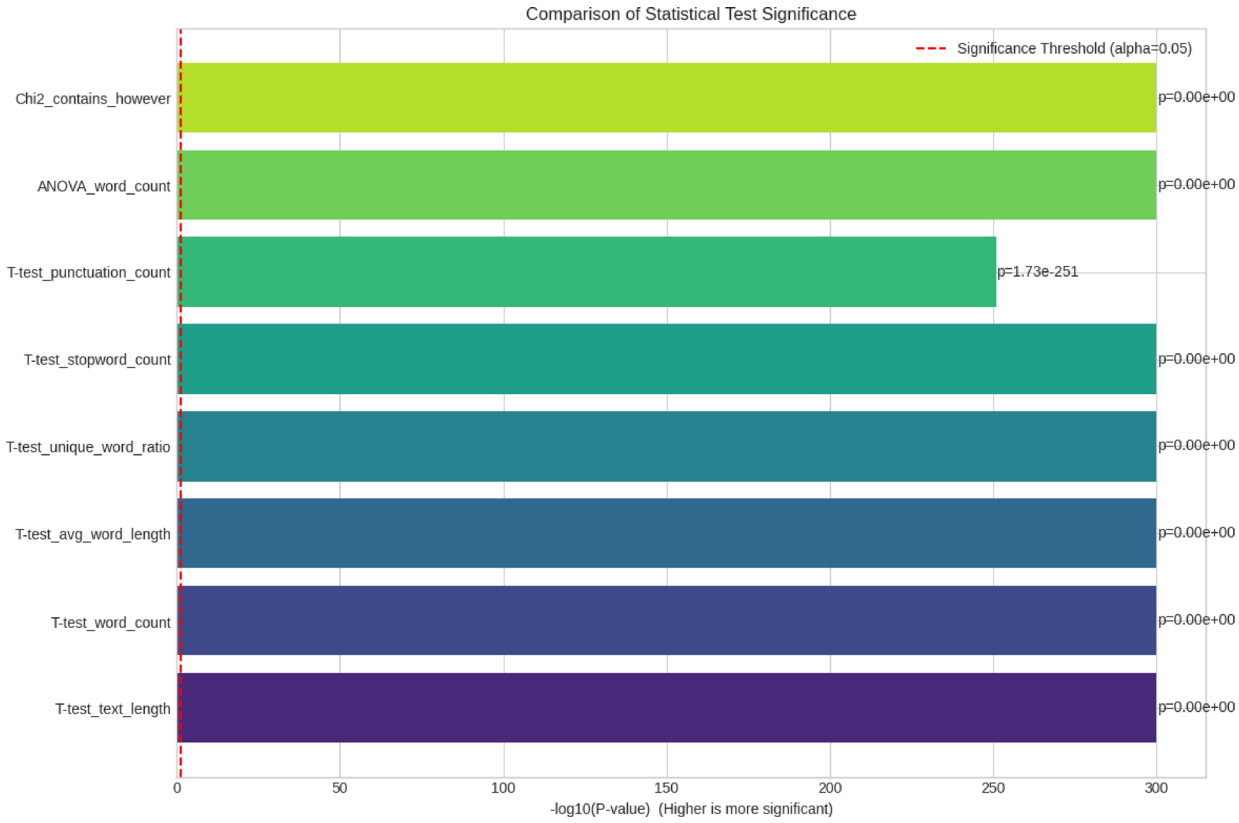

Fig. 18. Comparison of $-\log_{10}$(p-values) for statistical tests on test-set features, with the red dashed line marking $\alpha = 0.05$; this mixed display highlights which features (e.g., text length, punctuation count, unique-word ratio, "however" usage) are statistically significant discriminators and identifies non-significant attributes (e.g., average word length).

and assigns more importance to infrequent words in its attention heads. It is also noticeable that the punctuation count ANOVA ($p \approx 1.7 \times 10^{-251}$) shows that people use punctuation differently than AI, varying it for style and effect. DistilBERT's self-attention layers make use of punctuation tokens to improve the classification process.

This means that, like other discourse markers, the word "however" marks authorship differences between humans and AI. Even though it is less significant than the other metrics, its $-\log_{10}(p)$ (~ 4.6) still adds some support. In DistilBERT, this is reflected by small but persistent attention on discourse tokens, helping to reinforce the main signals. These statistically significant features show why the model achieved a high accuracy and F1 score, as well as which aspects of text are most and least helpful for AI-vs-human detection.

DistilBERT clearly outperforms traditional and deep learning models (such as RNN, LSTM and ensemble methods). It is an embodiment of its superiority due to its sophisticated contextual embedding and transfer learning capabilities to make best out of rich pretrained language representations trained on massive textual corpora. Its capacity to grasp deep semantic relationships, switch in context, and grammatical inferences, among other indicators of textual coherence, makes it apt at detecting stylistic or structural peculiarities inherent within AI-generated content, which to show, for the most part, superficial yet distinct types of reappearance, predictands, and demonstrably lower textual swing than human-produced content.

Thus, the strong predictive ability of DistilBERT, as indicated by its impressive accuracy and F1-score (0.98), indicated that it could be used as a better transformer model that is more robust and highly accurate in AI-text detection tasks. The results of this model also seem to have practical applicability in sensitive contexts, for example, in academic writing authenticity verification, real-time social media content validation, and cybersecurity against AI-built propaganda. As a result, these findings showcase the compelling reasons for utilizing sophisticated transformer approaches such as DistilBERT in identifying textual authenticity through distinguishing the authenticity of textual origins.

In the pursuit of identifying AI created from human-generated material, different models have been tested in this study as well as in a literature review. In this study, DistilBERT networks exhibited the highest performance, with an accuracy of 98%. Furthermore, it was noted that ensemble methods with GBM also achieved a maximum accuracy of 89% while revealing that these models were efficient at addressing this kind of classification problem. This performance corresponds with the results from prior works utilizing comparable deep learning techniques and ensembles for binary classification between AI and human writing, as shown in Fig. 19. Compared with the data in previous literature, where different types of neural network structures and boosting algorithms were used, the obtained outcomes in this paper validate the accuracy and application of both the LSTM and GBM

models for this classification task. This work contributes to the developing literature, which suggests that the techniques of deep learning and ensemble learning are particularly effective at dealing with the problem of AI-generated content identification with high accuracy.

Results using new corpus

The comparison with DistilBERT versus the original BERT model on the Human vs. LLM Text Corpus (used in prior work[34]) provides striking evidence of the efficiency and effectiveness of the distilled transformer model we propose. Fine-tuned BERT performance peaks at 90%, but even our simple pre-trained DistilBERT model with no specific feature engineering hits 95% performance—and absolute gain of 5%. This gain highlights the effectiveness of our targeted modifications to the transformer architecture, which involves incorporating subword-aware word embeddings as well as a customized self-attention head which actively amplifies stylistic and logical cues most indicative of AI-authored text. Accurcy, precision, recall and F1-score all gain proportionally with DistilBERT sustains all metrics at 0.95 on the held-out test set, whereas BERT reaches a performance plateau around 0.90. This benefit can again be illustrated as in Fig. 20 is the confusion matrix, where the number of false positives as well as false negatives have been dropped significantly. Only 53 "AI" samples were misclassified as human (much less than the BERT architecture) and 437 "humans" were misclassified as AI, suggesting that DistilBERT's attention mechanism is more optimally nuanced to distinguish the fine-grain patterns in machine generation. Consequently, our model not only pushes the current accuracy ceiling but also improves the consistency across all the classification metrics and hence able to make robust detection on different text.

Training and validation curves in Figs. 21 and 22 indicate that these improvements are a result of learning and not a consequence of overfitting, validation accuracy plateaus at 95% and loss steadily decreases, suggesting that DistilBERT quickly memorizes the salient discriminative features and generalizes across unseen examples. Early stopping had been performed after around the 40th epoch because the model's knowledge initialized pre-trained plus the fine-tuning followed by our class-weighted and augmentation strategies was quickly approaching an optimal solution. The combination of distilled architecture, subword-level and contextual embeddings, and adapted attention heads yields a lightweight yet highly accurate detector, capable of reliably identifying AI-generated text with minimal resource overhead. As demonstrated in the figures, the methodological advances yield clear performance gains and substantial improvements over state-of-the-arts, verifying the fundamental contributions of this work and establishing a new state-of-the-art baseline for AI-versus-human text detection.

Comparison with prior work

The proposed results are approximately 3% greater than those of previous works in terms of the accuracy of detecting AI-generated and human-generated content. The existing models, RNN[54], and RoBERTa[25], use conventional text analytics practices, including N-grams, POS taggers, and default feature representation. Here, these models achieved accuracies ranging from 80 to 91%. In contrast, using human written texts and BBC News datasets, more recent methods, such as DT[16], achieved accuracy of 89% using feature sets POS, Bigram and textual features. With the implementation of the state-of-the-art DistilBERT model on text data, the model outperformed existing studies, achieving 97% accuracy in detecting whether the text was generated by a bot or human. Additionally, LSTM with GloVe was embedded in the AIGC Content dataset; thus, the current existing models were improved with a high accuracy of approximately 93%, as displayed in Table 6. This significant performance improvement highlights the usefulness of establishing deep learning models such as LSTM and

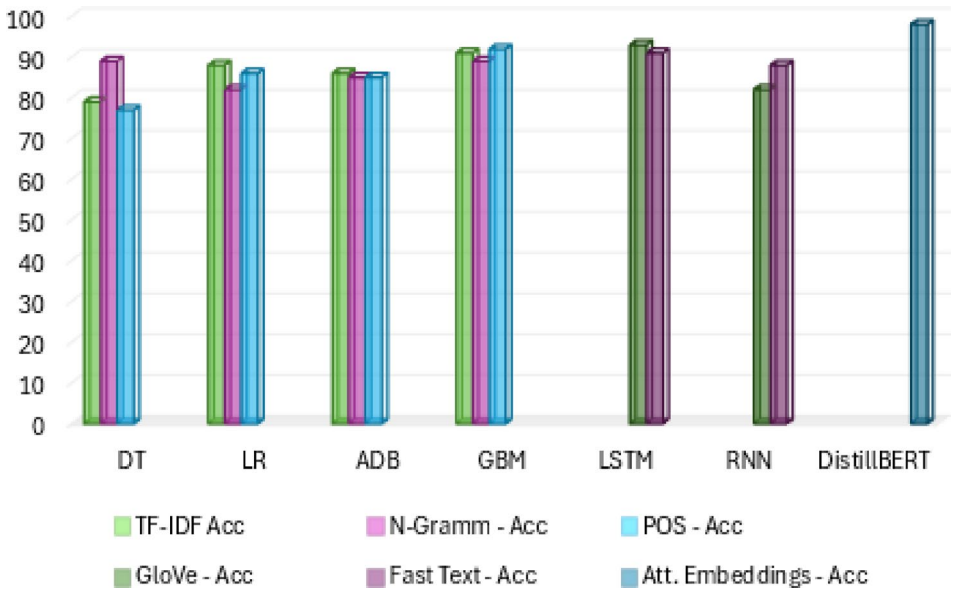

Fig. 19. Accuracy analysis of all applied models with features.

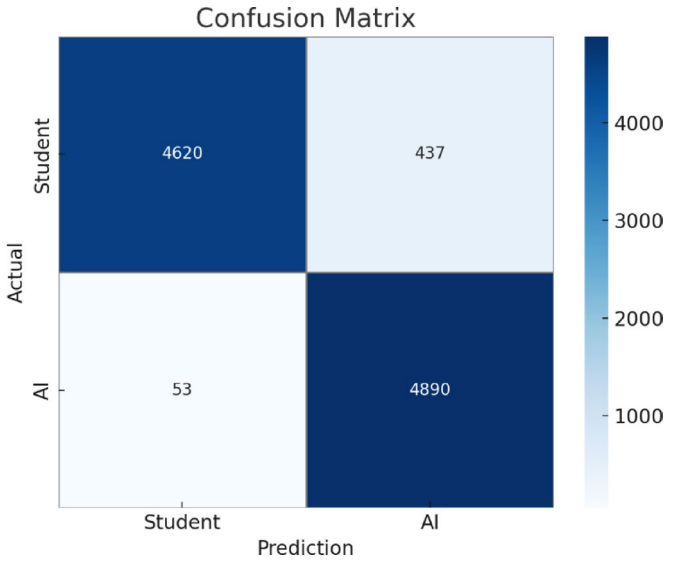

Fig. 20. Confusion matrix analysis based on identical dataset utilized in study.

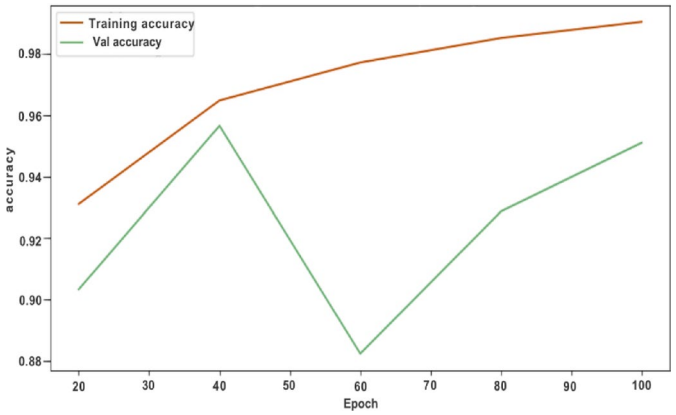

Fig. 21. Model training and validation accuracy analysis over epochs.

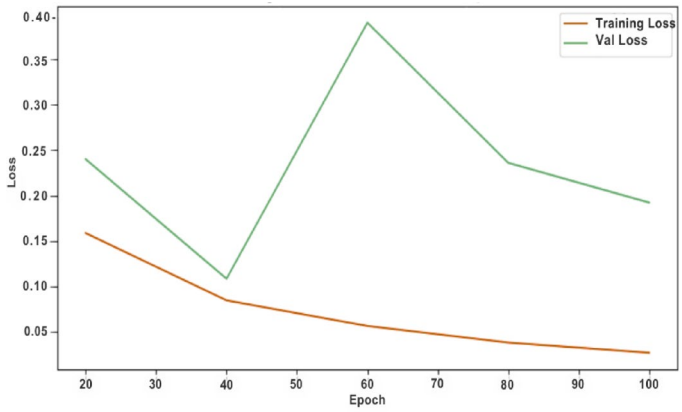

Fig. 22. Model training and validation loss analysis over epochs.

References	Year	Model	Dataset	Feature	Results
[54]	2022	RNN	Research paper content	POS	85
[25]	2022	RoBERTa	Tweets, Reddit comments, Yahoo answers, and Yelp user reviews. weets, Reddit comments, Yahoo answers, and Yelp user reviews. tweets, Reddit comments,	Default feature encoding	91
[16]	2024	DT	BBC News	Textual Features	89
[34]	2025	BERT	Essays	Pre-Trained Embeddings	90
Proposed	2025	Distil BERT	AIGC Content	Pretrained Embeddings	98

Table 6. Analysis of the proposed model with existing work (%).

DistilBERT in combination with better quality word embeddings for enhanced content classification. The improvement in the results can be attributed to the capacity of LSTM to model long-range dependencies in text data and the semantic representation from GloVe embedding.

Conclusion and future work

Advancement in AI has impacted various fields of human interaction and productivity, such as content making and dissemination. UGC significantly influences technology and has a massive impact on social media and the online community. As AI systems advance, the boundary between AI-generated and human-generated content is becoming increasingly blurred, leading to challenges in distinguishing the two. This has led to the emergence of a demanding necessity for effective identification systems to determine whether a given text was written by a human or an artificial intelligence (AI) model. This has been driven by the rapid advancement of AI in natural language processing (NLP), and the extensive adoption of applications such as chatbots, automated journalism or content creation has further compounded this issue. This research investigates the problem of distinguishing between generated text using both conventional machine learning and deep learning techniques. The results presented in this study show that the integration of the state-of-the-art transformer-based model DistilBERT achieved the highest accuracy—98%—with rich and high-quality word characteristics for the prediction of text nature. Our findings reveal that the combination of advanced DL techniques with high-quality word features, particularly using LSTM with a GloVe embedding accuracy of 93%, provides the most accurate results in distinguishing AI-generated from human-generated text, outperforming traditional ML models that rely on TF-IDF and POS-based features. Additionally, expanding the research by employing multimodal data such as image or audio data and behavioral data may provide a more comprehensive approach for differentiating between AI-generated content. Moreover, the recent improvements in the GPT and other large language models, including the GPT-4 and beyond, present the opportunity to strengthen detection methods and address the emerging issues posed by the increasing advancements in artificial intelligence (AI) systems. Future work can investigate the improvements in adversarial robustness with more advanced data augmentation, contrastive learning, and detection-evasion defense, to achieve robust performance under more advanced generative attacks. This research provides key insights into the potential to ensure transparency and trust in digital content, prevent misinformation and protect the credibility of user-generated content.

Data availability

The datasets generated and/or analyzed during the current study are available in the Kaggle repository, https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text; https://www.kaggle.com/datasets/starblasters8/human-vs-llm-text-corpus.

Received: 30 March 2025; Accepted: 19 June 2025
Published online: 01 July 2025

References

1. Li, Q., Zeng, Z., Li, T. & Sun, S. Identifying artificial intelligence–Generated content in online Q&A communities through interpretable machine learning. *J. Inf. Sci.*, 01655515241281491 (2024). https://doi.org/10.1177/01655515241281491.
2. Li, S. et al. Text mining of user-generated content (UGC) for business applications in E-commerce: A systematic review. *Mathematics* **10**(19), 3554. https://doi.org/10.3390/math10193554 (2022).
3. Shi, H., Dao, S. D. & Cai, J. LLMFormer: Large language model for open-vocabulary semantic segmentation. *Int. J. Comput. Vis.* **133**(2), 742–759. https://doi.org/10.1007/s11263-024-02171-y (2025).
4. Ishfaq, U., Khan, H. U. & Shabbir, D. Exploring the role of sentiment analysis with network and temporal features for finding influential users in social media platforms. *Soc. Netw. Anal. Min.* **14**(1), 241. https://doi.org/10.1007/s13278-024-01396-6 (2025).
5. Li, D. & Xing, W. A comparative study on sustainable development of online education platforms at home and abroad since the twenty-first century based on big data analysis. *Educ. Inf. Technol. (Dordr)* https://doi.org/10.1007/s10639-025-13400-3 (2025).
6. Katib, I., Assiri, F. Y., Abdushkour, H. A., Hamed, D. & Ragab, M. Differentiating chat generative pretrained transformer from humans: Detecting ChatGPT-generated text and human text using machine learning. *Mathematics* **11**(15), 3400. https://doi.org/10.3390/math11153400 (2023).
7. Wang, T. et al. Security and privacy on generative data in AIGC: A survey. *ACM Comput. Surv.* **57**(4), 1–34. https://doi.org/10.1145/3703626 (2024).
8. Pan, W. H. et al., Assessing AI detectors in identifying AI-generated code: implications for education. In *Proceedings of the 46th international conference on software engineering: software engineering education and training*, pp. 1–11 (ICSE-SEET '24. New York, NY, USA: Association for Computing Machinery, 2024). https://doi.org/10.1145/3639474.3640068.

9. Amirjalili, F., Neysani, M. & Nikbakht, A. Exploring the boundaries of authorship: A comparative analysis of AI-generated text and human academic writing in English literature. *Front. Educ. (Lausanne)*, vol. 9, (2024), https://doi.org/10.3389/feduc.2024.1347421.
10. Weber-Wulff, D. et al. Testing of detection tools for AI-generated text. *Int. J. Educ. Integr.* **19**(1), 26. https://doi.org/10.1007/s40979-023-00146-z (2023).
11. Naz, A., Khan, H. U., Alesawi, S., Abouola, O. I., Daud, A. & Ramzan, M. AI knows you: Deep learning model for prediction of extroversion personality trait. *IEEE Access*, 1, (2024), https://doi.org/10.1109/ACCESS.2024.3486578.
12. Khan, W., Daud, A., Khan, K., Muhammad, S. & Haq, R. Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Nat. Language Process. J.* **4**, 100026. https://doi.org/10.1016/j.nlp.2023.100026 (2023).
13. Ahmad, W., Khan, H. U., Iqbal, T. & Iqbal, S. Attention-based multi-channel gated recurrent neural networks: A novel feature-centric approach for aspect-based sentiment classification. *IEEE Access* **11**, 54408–54427. https://doi.org/10.1109/ACCESS.2023.3281889 (2023).
14. Cao, Y. et al. A survey of AI-generated content (AIGC). *ACM Comput. Surv.* **57**(5), 1–38. https://doi.org/10.1145/3704262 (2025).
15. Zaitsu, W. & Jin, M. Distinguishing ChatGPT (-3.5,-4)-generated and human-written papers through Japanese stylometric analysis. *PLoS ONE* **18**(8), e0288453. https://doi.org/10.1371/journal.pone.0288453 (2023).
16. Mathews, D., Varghese, J. P. & Samuel, L. C. Classifying AI-generated summaries and human summaries based on statistical features. In *2024 international conference on trends in quantum computing and emerging business technologies*, pp. 1–5 (2024). https://doi.org/10.1109/TQCEBT59414.2024.10545131.
17. Mitrović, S., Andreoletti, D. & Ayoub, O. ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated Text. Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.13852
18. Soni, M. & Wade, V. Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.17650
19. Tien, N. M. & Labbé, C. Detecting automatically generated sentences with grammatical structure similarity. *Scientometrics* **116**(2), 1247–1271. https://doi.org/10.1007/s11192-018-2789-4 (2018).
20. Cingillioglu, I. Detecting AI-generated essays: The ChatGPT challenge. *Int. J. Inf. Learn. Technol.* **40**(3), 259–268. https://doi.org/10.1108/IJILT-03-2023-0043 (2023).
21. Qian, K., Hu, B., Yamamoto, Y. & Schuller, B. W. The voice of the body: Why AI should listen to it and an archive. *Cyborg Bionic Syst.* **4**, 5. https://doi.org/10.34133/cbsystems.0005 (2023).
22. Stiff, H. & Johansson, F. Detecting computer-generated disinformation. *Int. J. Data Sci. Anal.* **13**(4), 363–383. https://doi.org/10.1007/s41060-021-00299-5 (2022).
23. Rodrigues, M., Silva, R., Borges, A. P., Franco, M. & Oliveira, C. Artificial intelligence: Threat or asset to academic integrity? A bibliometric analysis. *Kybernetes* **54**(5), 2939–2970. https://doi.org/10.1108/K-09-2023-1666 (2025).
24. Sajid, M., Sanaullah, M., Fuzail, M., Malik, T. S. & Shuhidan, S. M. Comparative analysis of text-based plagiarism detection techniques. *PLoS ONE* **20**(4), e0319551. https://doi.org/10.1371/journal.pone.0319551 (2025).
25. Arabi, H. & Akbari, M. Improving plagiarism detection in text document using hybrid weighted similarity. *Expert Syst. Appl.* **207**, 118034. https://doi.org/10.1016/j.eswa.2022.118034 (2022).
26. El-Rashidy, M. A., Mohamed, R. G., El-Fishawy, N. A. & Shouman, M. A. Reliable plagiarism detection system based on deep learning approaches. *Neural Comput. Appl.* **34**(21), 18837–18858. https://doi.org/10.1007/s00521-022-07486-w (2022).
27. Khalil, M. & Er, E. Will ChatGPT get you caught? Rethinking of plagiarism detection. In *Learning and collaboration technologies*, (A. Zaphiris Panayiotis and Ioannou, Ed.), pp. 475–487 (Cham: Springer Nature Switzerland, 2023).
28. Kayabas, A., Topcu, A. E., Alzoubi, Y. I. & Yıldız, M. A deep learning approach to classify AI-generated and human-written texts. *Appl. Sci.* **15**(10), 5541. https://doi.org/10.3390/app15105541 (2025).
29. 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. https://doi.org/10.1007/s00521-024-10538-y (2025).
30. Kumar, S., Tiwari, S., Prasad, R., Rana, A. & Arti, M. K. Comparative analysis of human and AI generated text. In *2024 11th international conference on signal processing and integrated networks (SPIN)*, pp. 168–173 (2024). https://doi.org/10.1109/SPIN60856.2024.10511301.
31. Tiwari, S., Sharma, R., Sikarwar, R. S., Dubey, G. P., Bajpai, N. & Singhatiya, S. Detecting AI generated content: A study of methods and applications. In *Proceedings of international conference on communication and computational technologies*, pp. 161–176 (Singapore: Springer Nature Singapore, 2024).
32. Sardinha, T. B. AI-generated vs human-authored texts: A multidimensional comparison. *Appl. Corpus Linguist.* **4**(1), 100083. https://doi.org/10.1016/j.acorp.2023.100083 (2024).
33. Boutadjine, A., Harrag, F. & Shaalan, K. Human vs. machine: A comparative study on the detection of AI-generated content. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, **24**(2), (2025), https://doi.org/10.1145/3708889.
34. Al Bataineh, A., Sickler, R., Kurcz, K. & Pedersen, K. AI-generated vs. human text: Introducing a new dataset for benchmarking and analysis. *IEEE Trans. Artif. Intell.*, pp. 1–11, (2025), https://doi.org/10.1109/TAI.2025.3544183.
35. Gui, J., Cui, B., Guo, X., Yu, K. & Wu, X. AIDER: A robust and topic-independent framework for detecting AI-generated text. In *Proceedings of the 31st international conference on computational linguistics*, (O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. Di Eugenio, and S. Schockaert, Eds.), pp. 9299–9310 (Abu Dhabi, UAE: Association for Computational Linguistics, 2025). [Online]. Available: https://aclanthology.org/2025.coling-main.625/
36. Blake, J., Miah, A. S. M., Kredens, K. & Shin, J. Detection of AI-generated texts: A Bi-LSTM and attention-based approach. *IEEE Access* **13**, 71563–71576. https://doi.org/10.1109/ACCESS.2025.3562750 (2025).
37. Aggarwal, K., Singh, S., Pal, V. & Yadav, S. S. A framework for enhancing accuracy in AI generated text detection using ensemble modelling. In *2024 IEEE region 10 symposium (TENSYMP)*, pp. 1–8 (2024). https://doi.org/10.1109/TENSYMP61132.2024.10752173.
38. Yu, D., Ai, J., Su, H. & Zhang, H. Assessing ChatGPT's comprehension of perturbed text through text linguistic features. In *2023 10th international conference on dependable systems and their applications (DSA)*, pp. 839–850 (2023). https://doi.org/10.1109/DSA59317.2023.00119.
39. Liu, B. et al. Adversarial attacks on large language model-based system and mitigating strategies: A case study on ChatGPT. *Secur. Commun. Netw.* **2023**(1), 8691095. https://doi.org/10.1155/2023/8691095 (2023).
40. Ye, D. et al. Optimizing AIGC services by prompt engineering and edge computing: A generative diffusion model-based contract theory approach. *IEEE Trans. Veh. Technol.* **74**(1), 571–586. https://doi.org/10.1109/TVT.2024.3463420 (2025).
41. Liu, Y. et al. Cross-modal generative semantic communications for mobile AIGC: Joint semantic encoding and prompt engineering. *IEEE Trans. Mob. Comput.* **23**(12), 14871–14888. https://doi.org/10.1109/TMC.2024.3449645 (2024).
42. Xu, W. Transformers-based feedback analysis of e-commerce: A focused study on quality assessment of agriculture products, (2025). [Online]. Available: http://creativecommons.org/licenses/by/4.0/
43. Muqadas, A. et al. Deep learning and sentence embeddings for detection of clickbait news from online content. *Sci. Rep.* **15**(1), 13251. https://doi.org/10.1038/s41598-025-97576-1 (2025).
44. Alsini, R. et al. Using deep learning and word embeddings for predicting human agreeableness behavior. *Sci. Rep.* **14**(1), 29875. https://doi.org/10.1038/s41598-024-81506-8 (2024).
45. Ding, J. et al. DialogueINAB: An interaction neural network based on attitudes and behaviors of interlocutors for dialogue emotion recognition. *J. Supercomput.* **79**(18), 20481–20514. https://doi.org/10.1007/s11227-023-05439-1 (2023).

46. Rojas-Simón, J., Ledeneva, Y. & García-Hernández, R. A. Classification of human and machine-generated texts using lexical features and supervised/unsupervised machine learning algorithms. In *Pattern Recognition*, (H. G. and C.-O. J. A. and M.-T. J. F. and O.-L. J. A. Mezura-Montes Efrén and Acosta-Mesa, Ed.), pp. 331–341 (Cham: Springer Nature Switzerland, 2024).
47. Ahmed, M., Khan, H. U., Iqbal, S. & Althebyan, Q. Automated question answering based on improved TF-IDF and cosine similarity. In *2022 ninth international conference on social networks analysis, management and security (SNAMS)*, pp. 1–6 (2022). https://doi.org/10.1109/SNAMS58071.2022.10062839.
48. Khan, W. et al. Part of speech tagging in Urdu: Comparison of machine and deep learning approaches. *IEEE Access* **7**, 38918–38936. https://doi.org/10.1109/ACCESS.2019.2897327 (2019).
49. Naqvi, S. M. M. R., Batool, S., Ahmed, M., Khan, H. U. & Shahid, M. A. A novel approach for building domain-specific chatbots by exploring sentence transformers-based encoding. In *2023 international conference on IT and industrial technologies (ICIT)*, pp. 1–7 (2023). https://doi.org/10.1109/ICIT59216.2023.10335884.
50. Albladi, A., Islam, M. & Seals, C. Sentiment analysis of twitter data using NLP models: A comprehensive review. *IEEE Access* **13**, 30444–30468. https://doi.org/10.1109/ACCESS.2025.3541494 (2025).
51. Terven, J., Cordova-Esparza, D.-M., Romero-González, J.-A., Ramírez-Pedraza, A. & Chávez-Urbiola, E. A. A comprehensive survey of loss functions and metrics in deep learning. *Artif. Intell. Rev.* **58**(7), 195. https://doi.org/10.1007/s10462-025-11198-7 (2025).
52. Naz, A. et al. Machine and deep learning for personality traits detection: A comprehensive survey and open research challenges. *Artif. Intell. Rev.* **58**(8), 239. https://doi.org/10.1007/s10462-025-11245-3 (2025).
53. Dang, V. M. H. & Verma, R. M. "Data quality in NLP: Metrics and a comprehensive taxonomy. In *Advances in intelligent data analysis XXII*, (N. and P. P. Miliou Ioanna and Piatkowski, Ed.), pp. 217–229 (Cham: Springer Nature Switzerland, 2024).
54. Najee-Ullah, A., Landeros, L., Balytskyi, Y. & Chang, S. Y. Towards detection of AI-generated texts and misinformation. In *Socio-technical aspects in security*, (L. Parkin Simon and Viganò, Ed.), pp. 194–205 (Cham: Springer International Publishing, 2022).

Acknowledgements

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. KFU252326].

Author contributions

All authors have contributed equally to this work.

Competing interest

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.U.K. or F.K.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025