National College of Ireland

# Detecting AI-Generated Text Using a Hybrid Deep Learning Framework for Authenticity Verification

MSc Research Project
Research Practicum

Ridhi Vipin Sharma
Student ID: 23297093

School of Computing
National College of Ireland

Supervisor: Cristina Hava Muntean

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Ridhi Vipin Sharma |
| **Student ID:** | 23297093 |
| **Programme:** | Msc in Data Analytics **Year:** 2025-2026 |
| **Module:** | Research Practicum |
| **Supervisor:** **Submission Due Date:** | January 28,2026 |
| **Project Title:** | Detecting AI-Generated Text Using a Hybrid Deep Learning Framework for Authenticity Verification |
| **Word Count:** | 7755 **Page Count** 21 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Ridhi Vipin Sharma

**Date:** January 27,2026

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Detecting AI-Generated Text Using a Hybrid Deep Learning Framework for Authenticity Verification

Ridhi Vipin Sharma
23297093

**Abstract**

The increasing prevalence of AI-generated text across digital platforms has created significant challenges for maintaining authenticity, preventing misinformation, and ensuring trustworthy communication. This study addresses the need for reliable detection of human-written and LLM-generated content by developing a lightweight hybrid architecture that integrates ALBERT's parameter-efficient contextual embeddings with the sequential learning capability of an LSTM network. Using a curated Human vs LLM dataset comprising text from Human, GPT-3.5, Claude-v1, and LLaMA-30B sources, the methodology follows the CRISP-DM framework, incorporating systematic data preparation, text cleaning, exploratory analysis, and model construction. ALBERT embeddings capture deep semantic and syntactic relationships within text, while the LSTM layer learns temporal and stylistic patterns characteristic of human and model-generated language. Evaluation using macro-average metrics demonstrates that the hybrid ALBERT+LSTM architecture achieves superior performance, obtaining an accuracy of **0.91** along with balanced precision, recall, and F1-score values of 0.91 across all classes. These results highlight the model's strong generalization capability and its ability to capture subtle linguistic distinctions even in short, informal social media posts. The study demonstrates the suitability of transformer–recurrent hybrid architectures for real-world text authenticity detection and emphasizes their potential for scalable deployment in environments where rapid and accurate identification of AI-generated content is essential.

# 1 Introduction

The recent breakneck development of artificial intelligence (AI) and natural language processing (NLP) has made large language models (LLM) like ChatGPT, LLaMA, and Claude write text closely similar to human writing. Such systems have transformed the digital communication, content and exchange of information. Nonetheless, they can replicate the patterns of human languages, which poses significant threats to online authenticity and integrity of information. Artificial intelligence-based content is being used more and more to spread fake news, control the discourse on social media, spam on a large scale and shape the opinion of the masses. On sites like Twitter, the semantic and contextual fluent nature of posts written by LLM makes them hard to tell the difference between actual expressions of a human being, posing a serious problem of content authentication and online credibility. Such a scenario points to the necessity of efficient and scalable systems that can identify AI-generated text in real-time digital contexts.

The number of AI-generated texts on the social media has grown significantly, as synthetic posts exhibit natural syntax, vocabulary that fits the situation, and stylistic uniformity. The latter features allow such content to evade the older detection mechanisms that use superficial lexical or statistical hints. The current machine learning and deep learning systems usually perform well on structured data but are not as effective and accurate in the case of noisy, multilingual, and dynamic social media environments (Chandana et al., 2024; Wani et al.,

2024). A hybrid system of detection combining transformer-based contextual understanding with sequence-modelling ability presents an appropriate path towards the identification of subtle linguistic patterns in short-form text. In this setting, an ALBERT+LSTM hybrid model incorporates the use of ALBERT to provide parameter-efficient contextual embeddings and LSTM to learn temporal and stylistic contextual dependencies.

The research is guided by the following central question: ***"How can a hybrid ALBERT+LSTM model be utilised to accurately and efficiently distinguish between human-written and LLM-generated tweets on social media?"***

It can be explained by the fact that the task of keeping real and trustful the online communication is getting harder and harder because developed language models create more and more content resembling human writing. Current tools to perform such detection, such as TF-IDF-based machine learning classifiers, or single deep learning models, lack contextual knowledge, have trouble with noisy short-form text, or are computationally expensive. Informal language, succinctness, and stylistic flexibility further complicate social media formats like Twitter, and standard models cannot be used to effectively detect them. These gaps are filled with a hybrid ALBERT + LSTM system that uses the parameter-efficient contextual embeddings of ALBERT and the ability of LSTM to learn sequential linguistic patterns, resulting in a system that can be used to encode nuances of semantic and temporal structure. Researching this hybrid approach is thus critical towards the creation of scalable, precise, and resource-saving solution to the real-time text authenticity detection in the contemporary digital world.

## Research Contribution

This study contributes to the field of AI-generated text detection by developing and evaluating a lightweight hybrid detection architecture designed for short-form, informal social media content. The primary contributions are:

1. Development of a hybrid ALBERT+LSTM architecture that combines parameter-efficient contextual embeddings with sequential modelling for authenticity classification.
2. Comparative evaluation of the hybrid architecture against common baseline machine learning models to assess relative performance and computational efficiency.
3. Demonstration of enhanced generalisation capability on short, noisy, and informal text commonly found on platforms such as Twitter.
4. Identification of current limitations within existing detection approaches, establishing the necessity for hybrid methods that balance contextual depth with efficiency.

The objectives of this study are:

- To design a hybrid ALBERT+LSTM architecture capable of capturing contextual and sequential linguistic features relevant to classification.

- To prepare and analyse a human-versus-LLM text dataset through systematic pre-processing and feature exploration.
- To train and evaluate the hybrid architecture using metrics such as accuracy, precision, recall, and F1-score.
- To compare classification performance against baseline machine learning classifiers under consistent experimental conditions.

# 2 Related Work

The studies on differentiating between human and AI-generated text have increased at a fast pace with the development of sophisticated large language models (LLMs). The current solutions are based on either conventional machine learning (ML) methods or deep learning (DL) models and systems based on transformers. In this section, the most pertinent studies have been critically analyzed in terms of methodologies, datasets, performance outcomes, and limitations.

## 2.1 Machine Learning Approaches for Text Authenticity Detection

Initial studies on text authenticity were based on classical ML classifiers with manually designed linguistic features. Elmogy et al. (2021) compared fake review identification with such supervised models as KNN, Naive Bayes, SVM, logistic regression, and Random Forest. The paper has shown that the integration of textual and reviewer behaviour characteristics enhanced the classification results. Nevertheless, the study used mostly structured review datasets, and did not use semantic embeddings. The models relied on shallow lexical information, which restricted their capacity to identify hidden patterns that exist in advanced AI-generated texts. Also, generalisation was limited to informal, short social media posts because of the lack of contextual modelling. Alsubari et al. (2022) further developed ML-based detection by including TF-IDF representations and ensemble techniques and achieved high accuracy (95) in detecting manipulated hotel reviews. Although the findings may depict the usefulness of the n-gram based features in the process of capturing stylistic abnormalities, the method is susceptible to vocabulary anomalies and domain shifts. The method is not semantically abundant and is not capable of distinguishing well-constructed content produced by LLM, which is more likely to resemble human syntax than older false reviews.

Yang (2025) compared product reviews created by AI on Amazon through a TF-IDF + SVM pipeline which attained a 99.25% F1-score. The model is also highly performing, though it is based on sparse lexical features that cannot be easily generalised to noisy data sets such as Twitter. The research is also concerned with longer and structured reviews instead of short-form texts, which restricts its applicability to the platforms where linguistic brevity and variability prevail. Sadiq et al. (2023) proposed a hybrid ML-DL architecture in which the FastText embeddings and CNN classifier are used to detect deep fake tweets. The model came up with 93% accuracy on the TweepFake dataset which is a better contextual awareness compared to TF-IDF models. Nevertheless, FastText embeddings represent a small number

of long-range dependencies and the CNN model does not process sequential information, which prevents it from distinguishing subtle stylistic clues that are often present in LLM-generated text.

**Critical Analysis:**

Together, ML-based approaches are computationally efficient and interpretable but do not have the ability to model semantic dependencies and sequence-level patterns. They cannot be flexible to a wide range of writing styles and even highly developed AI-generated results due to their dependence on manual functionality. These constraints explain why it is necessary to have architectures with contextual embeddings and temporal modelling.

## 2.2 Deep Learning Architectures for AI-Generated Text Detection

Text authenticity detection has been developed through deep learning by learning hierarchical features directly on the raw text. Recently, Fagni et al. (2021) presented the TweepFake dataset and tested the models, including RNNs and LSTMs and GPT-2 classifiers. The work provided a standard on deep fake tweet detection; nonetheless, various models had a low efficiency on informal or noisy text and previous transformer versions were not efficient enough to use in real-time. The CNN-BiLSTM architecture used by Bhuvaneshwari et al. (2021) was improved with the self-attention mechanism, which allowed the model to provide an importance weight to the important tokens. Although this hybrid structure enhanced interpretability and contextual learning, it was still computationally expensive and was optimised to structured review data and not the short posts on social media. Kayabas et al. (2025) used an LSTM network to determine whether a Turkish text is written by a human or an AI and obtained a high F1-score of 0.97. Despite the strength of the results obtained with the help of sequential modelling, the method is based exclusively on LSTM and transformer-based contextual embeddings. This restricts the performance on the datasets with complicated semantic variations.

Blake et al. (2025) presented a BiLSTM with attention and residual links with the highest accuracy of 88%. The architecture was shown to be more focused on linguistically relevant patterns but consumed a lot of memory, making it less suitable to be applied to large scale, real-time detection pipelines. A comparison of CNN, LSTM, GRU, and hybrid CNN-LSTM architectures by Chandana et al. (2024) revealed that sequential-spatial learning increased the accuracy of AI-generated text detection. Nevertheless, the models had long training durations, were not optimised towards scalability and their performance was inconsistent with noisy datasets. Wani et al. (2024) and Ghatora et al. (2024) investigated the use of DL models together with pre-trained embeddings (Word2Vec, GPT-4), showing better semantic accuracy. However, such methods also needed a large amount of computing power and were less efficient over large text streams.

**Critical Analysis:**

DL architectures have good capabilities of language context modelling however their computational costs and training requirements present a limitation to real world tasks. Most of the models rely on intensive attention models or are inefficient in terms of parameter, preventing use in high-speed applications like Twitter. Moreover, detectors based on transformers are frequently opaque black-box systems, which makes them hard to interpret.

## 2.3 Advanced Hybrid and Transformer-Based Approaches

Recent studies have been directed towards improving detection using hybrid or transformer-based architectures. The model of stylistic analysis developed by Rosca et al. (2025) is characterized by almost perfect accuracy (as high as 99) based on the handcrafted linguistic features. Although the performance was good, the system had a lot of feature engineering and was not able to scale across domains. Hireche et al. (2025) used Bi-GRUs with linguistic features like perplexity, readability, and syntactic complexity, attaining a high accuracy of 98% on the data obtained using DAIGT. The architecture despite the high performance showed a higher training time and lower generalisability over informal text. Mitrovic et al. (2023) applied explainable AI to find the distinguishing features in ChatGPT-generated text but observed a decrease in accuracy on rephrased or semantically shifted data (79% accuracy). The paper has shown shortcomings of transformer-only detectors in the presence of context variation.

**Critical Analysis:**

Despite the good performance of hybrid and transformer-based methods, most of the examined architectures are resource intensive and do not focus on efficiency. They are limited in the scalability of continuous and real-time social media monitoring due to their reliance on big parameter sets or manually crafted features.

**Table 1. Summary Table of Related Works**

| Author(s) & Year | Work Description | Model Used | Key Contribution | Limitations |
|---|---|---|---|---|
| **Elmogy et al. (2021)** | Fake review detection on Yelp dataset using text and behavioural features. | KNN, NB, SVM, RF | Demonstrated importance of combining reviewer behaviour with textual features. | ML models fail to capture semantic dependencies; limited contextual understanding. |
| **Alsubari et al. (2022)** | Identification of hotel reviews via text n-grams and sentiment analysis. | NB, SVM, RF, AB | High classification accuracy with TF–IDF and | Relies on static features; lacks contextual embeddings and |

| | | | | |
|---|---|---|---|---|
| | | | sentiment-based features. | sequential learning. |
| **Fagni et al. (2021)** | Creation of TweepFake dataset and baseline evaluations for fake tweet detection. | RNN, LSTM, GPT-2, ML baselines | First public dataset enabling deep fake text research on Twitter. | Benchmark limited to older models; does not exploit lightweight transformers. |
| **Yang (2025)** | AI-generated product review classification on Amazon. | TF–IDF + SVM | Achieved 99.25% F1-score with simple feature–classifier pipeline. | High accuracy but no semantic embeddings; poor generalization to informal tweets. |
| **Bhuvaneshwari et al. (2021)** | Spam review detection via CNN–BiLSTM with self-attention. | CNN + BiLSTM | Improved contextual learning through hybrid DL architecture. | Computationally heavy; lacks efficiency for short social media texts. |
| **Kayabas et al. (2025)** | AI-generated text detection in Turkish using LSTM. | LSTM | Achieved 97% accuracy with efficient sequence modelling. | Focused on monolingual dataset; lacks transformer-level contextuality. |
| **Chandana et al. (2024)** | Detection of AI-generated text using multiple DL models. | CNN, LSTM, GRU, CNN+LSTM | Demonstrated comparative performance across deep networks. | Models are large and require long training; limited optimization for scalability. |
| **Rosca et al. (2025)** | Deep fake text identification via linguistic pattern recognition. | Custom linguistic ML model | Captures individual writing style and author-specific text features. | Requires handcrafted features and heavy computation; not scalable. |
| **Ghatora et al. (2024)** | Sentiment analysis using traditional ML and LLMs (e.g., GPT-4). | SVM, RF, GPT-4 | Highlighted superiority of LLMs in contextual sentiment analysis. | LLMs demand high resources; impractical for lightweight detection systems. |

| | | | | |
|---|---|---|---|---|
| **Ahammad et al. (2024)** | Sentiment analysis of AI app reviews with multiple DL methods. | CNN, LSTM, GRU, BiGRU | Found CNN+LSTM achieves highest accuracy (92.1%). | Performance limited to structured review datasets; lacks adaptability to short texts. |

## 2.4 Research Gaps and Motivation

Even though a great number of improvements have been made, a review of the literature shows that a number of gaps in research remain in the area of deep fake text detection. Most ML methods have been based on manual features that do not transfer between different platforms and languages (Elmogy et al., 2021; Alsubari et al., 2022). Even powerful deep learning models may be susceptible to overfitting in cases where training data is small or inaccurate (Fagni et al., 2021; Bhuvaneshwari et al., 2021). Models based on transformers, despite having better contextual knowledge, add too much computational cost, latency, and memory (Hireche et al., 2025; Rosca et al., 2025). Furthermore, it is not easy to explain and the models such as GPT-based detectors are frequently viewed as black boxes (Mitrovic et al., 2023). Little is also done to explore lightweight, end-to-end models that are tailored to short, noisy social media texts, particularly twitter, where linguistic conciseness and informal writing make semantic analysis challenging (Sadiq et al., 2023).

The ALBERT+LSTM model is good at overcoming these shortcomings. ALBERT offers parameter-efficient contextual embeddings, which significantly decrease the memory consumption but do not sacrifice semantic richness. With a layer of ALBERT and an LSTM, it is possible to learn sequential dependencies and the model is able to learn the contextual and temporal relationship between tokens, which is critical to distinguishing slight differences between human- and AI-generated tweets. A balanced trade-off between the interpretability, scalability and performance is achieved with the pipeline, which includes cleaning the text with tokens, embedding using ALBERT, and processing with LSTM and dense-SoftMax classification. This architecture is efficient, robust, and also exhibits high detection accuracy by integrating both transformer-based contextual representation and the sequential modeling ability of LSTM. It is therefore a good trade-off between rich contextual learning and thin computational design and therefore it is very much applicable in the real-world social media usage.

# 3 Research Methodology

The research methodology that will be used in the study is a systematic series of actions planned to facilitate the identification of human-written and LLM-generated tweets with the assistance of a hybrid ALBERT+LSTM architecture. The general methodological process comprises the acquisition of the dataset, its validation, text preprocessing, exploratory data analysis, feature extraction, the development of the base model, the specification of the

hybrid architecture, and evaluation. The workflow adheres to the accepted data analytics research processes and is clear, reproducible, and methodologically sound.

## 3.1 CRISP-DM Framework

The research design is based on the Cross-Industry Standard Process of Data Mining (CRISP-DM), an analytical framework that is popular and offers a systematic guide to the creation of solutions based on data. CRISP-DM involves six iterative steps, which are business understanding, data understanding, data preparation, modelling, evaluation, and deployment. In this paper, the framework will be used to systematically change raw text into a predictive classification system by initially defining the problem situation, studying the properties of the dataset, and organizing the textual data by cleaning and normalizing it. The next steps are concerned with building baseline models and the hybrid ALBERT+LSTM architecture development, and the evaluation of the performance based on the defined metrics. CRISP-DM usage guarantees a methodological transparency, reproducibility, and correspondence to the best practices in data analytics studies.

## 3.2 Dataset Description

The dataset was obtained from the publicly available *Human vs LLM Text Corpus* published on Kaggle https://www.kaggle.com/datasets/starblasters8/human-vs-llm-text-corpus/data, which includes labelled samples from Human, GPT-3.5, Claude-v1, and LLaMA. The dataset was in the form of 31, 180 rows and 5 columns, with samples of text and metadata thereof. The dataset is a filtered set that aims at the linguistic variations between human-written and AI-generated text. The dataset is a collection of diverse text samples that are obtained in various areas, such as social media, online conversations, and generative language models. The records give organized data in the form of raw text content, the source label of the text, whether it was written by a human or an AI model, and auxiliary data in the form of prompt identifiers, text length, and word count. The data set is very diverse in terms of tones in writing, the complexity of the sentences and contextual expression, which is a real-life variance in the form of communication. It contains short and informal sentences as well as longer and context-sensitive responses, which makes it appropriate to analyse stylistic and semantic patterns that distinguish human language and language models generated by a large language model. Such diversity makes the dataset representative, balanced, and linguistically diverse, which provides a solid basis to continued analysis and develop models.

## 3.3 Data Pre-processing

Pre-processing of the data was conducted in order to make the corpus standard, eliminate textual noise, and prepare the data to be utilized in feature extraction and model training. The procedure used in this is as follows:

The data was initially checked on the missing values, inconsistency of records and duplication, as there were no nulls, the duplicates were eliminated to uphold integrity of data and avoid biased learning. Then, the source attribute was checked with exploratory checks

with descriptive statistics and bar charts to make sure that the balance of classes, both human and AI-created samples, was appropriate. To perform a normalization of the texts, the text column was first normalised through NLTK: all text was lowercased, all URLs, numbers and punctuations were eliminated through regular expressions, the cleaned text was tokenised with a word tokeniser of NLTK. Stops in English and extremely short tokens (less than three characters) were then eliminated to minimize noise and lemmatization was used with WordNet to transform words to their base forms to enhance semantic consistency and eliminate redundancies. Post-processing Tokens were reassembled into clean text strings and stored in a new cleantext column after pre-processing, and a derived feature, cleanword_count, was calculated to record the length of each cleaned sample. Lastly, histograms and Kernel Density Estimation (KDE) plots were created to see the distribution of cleaned word counts and to measure the variability of text-length, no capping or elimination of outliers was done to preserve the natural variability between human and AI-generated samples. A stratified split with 80:20 split was used to split the dataset to 18,687 samples in the training set and 4,672 samples in the test set.

## 3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was done to analyze the structure, quality and language of the dataset before the development of the model. The data set was initially verified to be complete and consistent and there were no missing/ null values, which showed good data quality. The same data were then filtered out to eliminate any duplicate records and so the same samples were not used in training and evaluating the model and a total of 7,821 duplicates were eliminated out of 31,180 records, and a total of 23,359 distinct text samples were obtained. The source distribution was then examined through frequency counts and bar charts to provide a fair balance between human-written and AI-generated text where the majority of the samples were found in Human and GPT-3.5 sources and smaller proportions in models such as LLaMA-30B and Claude-v1 which can be used to draw unbiased comparison of classifications. Descriptive statistics and clean word count visualization (histogram + KDE) were used to study text-length behavior with a distribution that is predominantly short-form where the concentration is strong at lower word counts and long-tailed to the right, where there are fewer very long texts. More precisely, approximately 72 percent of samples have less than 100 words, 22 percent of samples have 100-300 words and only 6 percent have over 300 words which are typical of short conversational or social-media-style content.

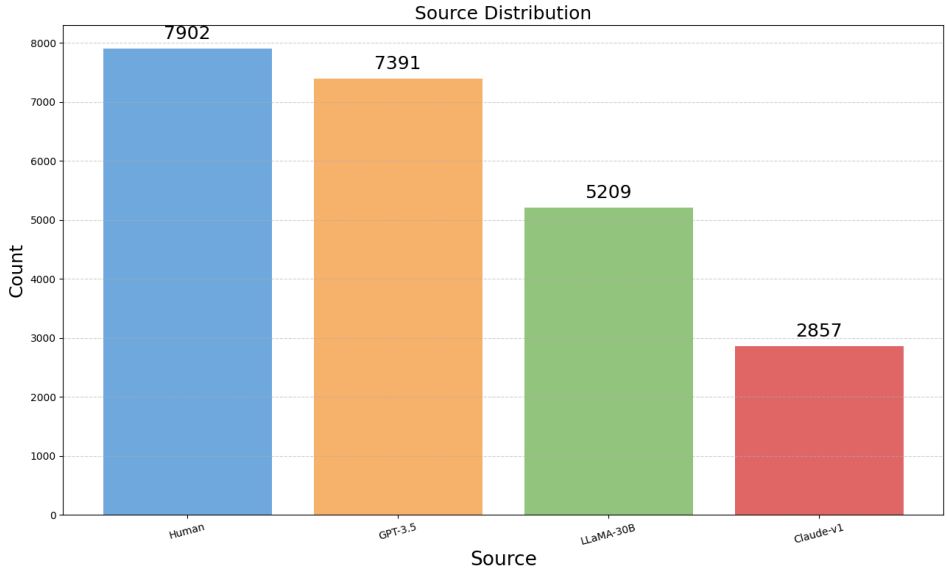


**Figure 1. Source Distribution of Human and LLM-Generated Texts**

In Figure 1, the bar chart shows the distribution of the text samples in various sources, such as the text written by humans and the text generated by various large language models (LLMs). Human and GPT-3.5 texts prevail with a small number of samples of LLaMA-30B and Claude-v1. This visualization aids in knowing the relative representation of each source in the dataset, so that the analysis will be balanced when assessing the model as well as comparing them.

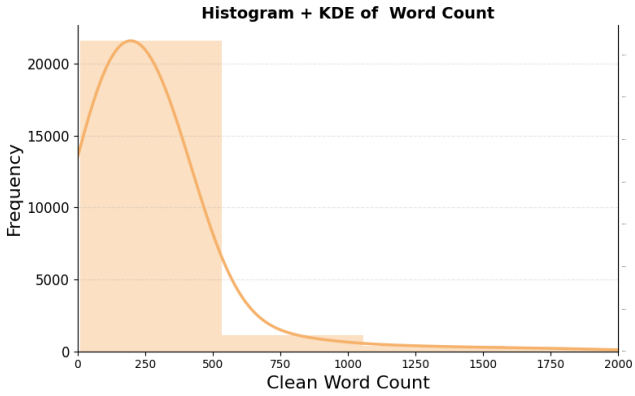


**Figure 2. Histogram and KDE of Clean Word Count**

Figure 2 gives a plot of a histogram and a kernel density estimate (KDE) to visualize the distribution of the number of clean words in the dataset. The steep slope around the low-end range points to the fact that the majority of text samples are comparably small with fewer words. The long right tail indicates that there are some few longer texts, which is a variability in the length of the text across the sources.


**Figure 3. Word Cloud Comparison of Human and LLM-Generated Texts**

Figure 3 is a visualization that makes a comparison between the most commonly used words in texts written by humans and different large language models (LLMs). The subplots are the sources of different size, but the frequency of the words is reflected in the size of the subplot.

## 3.5 Feature Extraction Using TF–IDF

The clean corpus was processed into numerical features representations based on the **Term Frequency-Inverse Document Frequency (TF-IDF)** vectorization technique after text pre-processing. TF-IDF measures the significance of each word in comparison to its occurrence in a document, and the number of times that word appears in the corpus, allowing the model to give importance to meaningful words and deemphasize common or non-informative words. The process of vectorization was programmed in a way that the most striking unigrams and bigrams could be extracted, both individual words and short word combinations that bring about stylistic and contextual dissimilarities between human-generated and AI-generated text. Stopwords were also removed to reduce redundancy and the vocabulary size was also kept to a fixed number of features to ensure the computational efficiency. This representation transformed the text cleaned into a sparse matrix format with each row being a document and each column a term feature. The resulting feature matrix was used as the input to the baseline classification models, which offered a trade-off between interpretability and expressiveness to the analysis of linguistic patterns.

## 3.6 Model Selection and Rationale

The choice of the hybrid ALBERT+LSTM structure is justified by the fact that a detection system is required that will be able to recognize both contextual semantics and sequential linguistic patterns of the human-written and the text created by the LLM. Even the classical models of machine learning are efficient, but mostly rely on sparse lexical representations like TF-IDF and thus cannot capture deep contextual relationships. Transformer-based models have good semantic insights and can be computationally intensive, thus not applicable to large scale or real time applications. ALBERT (A Lite BERT) mitigates these limitations by using parameter reduction methods, such as factorized embedding

parameterization and cross-layer weight sharing, which substantially decrease the memory use and still maintain the contextual richness. The embeddings of transformers, however, do not explicitly represent the temporal dependence, stylistic development, or sequence behaviour at the level of tokens, which are the major indicators that distinguish human linguistic patterns and those of LLM output. This can be achieved by incorporating ALBERT embeddings to an LSTM layer in order to have the architecture combine both the strong semantic encoding with long-range sequence modelling to allow the system to pick up fine structural information that conventional models often lack. This trade-off is consistent with the goal of high classification accuracy with low computational cost, interpretability and appropriateness to short informal social media text (tweets).

## 3.7 ALBERT+LSTM Architecture

The ALBERT+LSTM model combines contextual transformer-based embeddings, which are combined with sequential neural processing to improve text authenticity detection. ALBERT (A Lite BERT) is the embedding part, which takes raw text as input to generate dense contextual representations using its factorized embedding parameterization strategy and cross-layer weight sharing strategies. These design features make the models much smaller and less memory intensive and still allow modeling of the semantic relationships, syntactic structure, and word-level dependencies in the input text. ALBERT creates a train of contextual embeddings such that each token is modeled as part of a context, allowing the architecture to capture finer linguistic details that make human-written text and text by LLM-generated text distinct.

The contextual embeddings generated by ALBERT are then sent into a Long Short-Term Memory (LSTM) network which is specialised in the model of temporal dependencies and sequence patterns. The LSTM layer processes tokens sequentially where it learns stylistic transition, coherence patterns and temporal patterns typical of human language behaviour. This sequential analysis allows the detection system to obtain features that may not be completely differentiated by embeddings of transformers. The output of the LSTM is fed to dense layers and SoftMax classifier which gives the probability scores of each class, and produces a final prediction on whether the text is a human or an LLM. ALBERT and LSTM have a combination of contextual depth and temporal modelling that make the architecture both computationally and highly effective in short, informal, and stylistically diverse social media text.

## 3.8 Evaluation Strategy

The evaluation plan will determine the effectiveness of the hybrid ALBERT+LSTM architecture in the typical text classification measures. The data is stratified into a training and testing dataset to maintain the distribution of classes and the model performance is evaluated by accuracy, precision, recall and F1-score. All multi-class metrics are evaluated on macro-average values to make sure that every class has an equal contribution to the assessment regardless of the imbalance in the classes. Misclassification patterns are analysed

using confusion matrices to determine systematic errors between categories. The loss curves of training and validation are analyzed to ensure that the convergence is steady and there is no overfitting. A comparative analysis with the machine learning and deep learning baselines offers further background information on efficiency and increase in classification ability.

# 4 Design Specification

## 4.1 System Architecture Overview

The system architecture is designed to be based on a multi-stage analytical pipeline that will be used to classify text as either human-written or LLM-generated. The hierarchy adopted in the architecture is that of data collection and then pre-processing, modelling and evaluation. The stages are described as autonomous but interdependent modules, which allows transparency, reproducibility, and separation of functionality. Three branches are found in the modelling block, including machine learning models, deep learning models, and the hybrid architecture ALBERT+LSTM. The parallel design enables the comparison of performance of various modelling families and therefore the system can determine the best technique that is used in text authenticity detection. It has a modular architecture that can be easily extended with different algorithms without changing the workflow.

The hybrid ALBERT+LSTM architecture takes the middle position in the system because it has the capacity to integrate contextual transformer embedding with sequential learning. Machine learning algorithms like Logistic Regression and SVM use TF-IDF representations and they offer effective baselines whereas deep learning models like GRU or LSTM can offer increased ability to capture sequential patterns. The evaluation module summarizes all the modelling branches and calculates the accuracy, precision, recall, and F1-score to identify the performance of classification. The systematic design will provide uniformity in model outputs and will allow systematic comparison, hence making it possible to have a trusted evaluation of the appropriateness of every modelling approach.

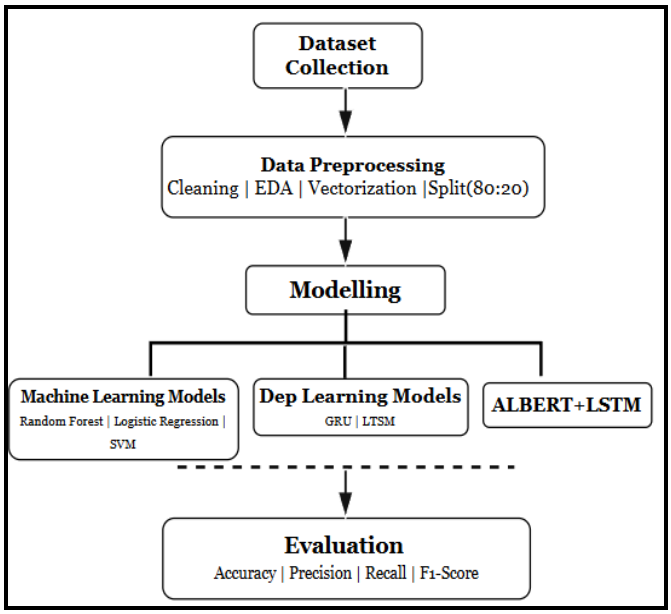

**Figure 4. Human vs LLM Text Classification End-to-End.**

The figure 4 demonstrates the overall workflow of the study, including how the dataset is moving through the pre-processing and various modelling methods.

## 4.2 Workflow Pipeline

The workflow pipeline is a series of steps starting with the acquisition of data to the analysis of the model, and it is organized in the order depicted in the architecture diagram. It starts with collecting data samples by gathering human-generated and LLM-generated texts and preparing them to be analyzed. Data pre-processing step involves cleaning, normalizing, exploratory data analysis, vectorisation (where necessary) and 80:20 train-test split. These processes normalise the textual inputs and produce both the raw and derived features that are needed by the next modelling elements. Pre-processing ensures that noise is eliminated, linguistic structure is maintained and that the dataset is formatted in machine learning, deep learning and hybrid architecture format.

After pre-processing, the pipeline has three parallel routes of modelling. Algorithms used in the machine learning pathway include Logistic Regression and SVM with TF-IDF features, whereas the deep learning pathway involves the use of GRU or LSTM networks to analyse sequential patterns. The hybrid ALBERT+LSTM model is a combination of transformer-based embeddings and temporal learning mechanisms and serves as the core analysis framework. The results of all modelling paths are summarized in the evaluation phase during which accuracy, precision, recall, and F1-score are calculated to evaluate the performance of the model. The workflow design allows the thorough comparative analysis of the modelling families and makes the final outcomes indicate the predictive power and the methodological strength of the results.

## 4.3 Libraries and Tools Used

The code of the system implementation is based on Python and a collection of special libraries to process, model, and evaluate data. NumPy and Pandas assist in structuring of data and preprocessing, whereas NLTK allows to normalize text using tokenization, stop word elimination, and lemmatization. Scikit-learn is used to implement machine learning models and TF-IDF vectorization, and sequence modelling is done by deep learning with the help of TensorFlow and Keras. The hybrid ALBERT+LSTM model is a system that uses the Hugging Face Transformers library to generate contextual embeddings. Exploratory data analysis and performance assessment visualizations are made with Matplotlib and Seaborn.

# 5 Implementation

## 5.1 Machine Learning Models Implementation

The machine learning application will involve three baseline classifiers, namely, Random Forest, Logistic Regression, and Linear Support Vector Machine (SVM), each of which will be optimised by systematic hyperparameter tuning. In the case of Random Forest, the number of estimators is also considered in multiple values of 100 to 1000 to determine the number of trees that offer the highest cross-validation accuracy to balance the size of the ensemble with learning stability. Logistic Regression is optimised by varying C parameter, whereby the range of values between 0.01 and 10 were tried to identify the regularisation force that provides the most optimal generalisation performance on high-dimensional TF-IDF features. The same tuning strategy is used in the Linear SVM model where the same range of C is searched to determine the best regularisation level to use to construct a solid decision boundary.

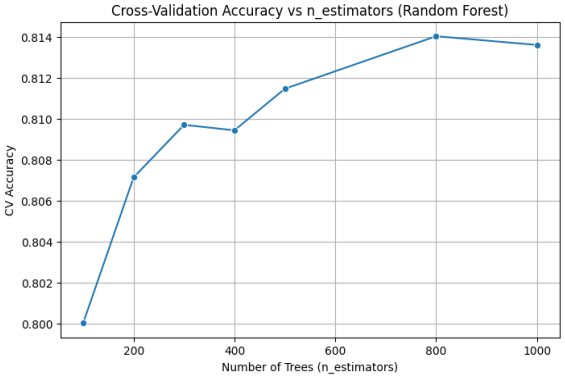


**Figure 5. Cross-Validation Accuracy vs Number of Trees**

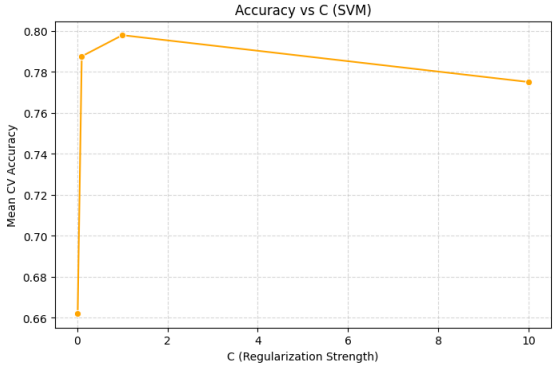


**Figure 6. Accuracy vs Regularisation Strength**

Figure 5 and Figure 6 plots indicate that model performance is enhanced when significant hyperparameters are systematically tuned, which explains the importance of parameter optimisation in enhancing better accuracy.

## 5.2 Deep Learning Models Implementation

The implementation of deep learning is made up of two sequence-based models, a Bidirectional GRU model and a Bidirectional LSTM model, which are trained to learn temporal and contextual patterns in text. Both models use a vocabulary-limited embedding layer with 60,000 tokens and 128-dimensional embedding space that uses dense vector representations of input sequences. Both architectures use a maximum length of sequence of 400 tokens to provide standardised input and padding and truncation are done in all samples. The Bidirectional GRU model uses a recurrent layer of 128 units that are configured to produce sequential data and a global max-pooling operation that combines the information of all the time-steps. The learned representation is refined by a fully connected dense layer with

128 units and the final SoftMax layer generates class probabilities of human-generated or LLM-generated text.

The Bidirectional LSTM model adopts a similar structural configuration but replaces the GRU layer with a 128-unit bidirectional LSTM layer, enabling the extraction of longer-range dependencies and more complex sequential relationships within the text. Both models are trained using the Adam optimiser and the sparse categorical cross-entropy loss function, which are widely recognised for their stability and suitability in multi-class classification tasks. A validation split of 10% is applied during training to monitor generalisation behaviour, and early stopping is implemented based on validation loss to prevent unnecessary training and reduce the risk of overfitting. Batch size, sequence length, embedding dimensions, vocabulary size, and recurrent unit count are kept consistent across both architectures to support a fair comparison of modelling capability between GRU- and LSTM-based designs.

## 5.3 Hybrid ALBERT+LSTM Implementation

The hybrid ALBERT+LSTM implementation combines transformer-based contextual embeddings with sequential neural processing to enhance the differentiation between human-written and LLM-generated text. The process begins with label encoding and dataset partitioning using an 80:20 stratified split to preserve class balance. Text inputs are tokenised using the *albert-base-v2* tokenizer with padding and truncation set to a maximum length of 256 tokens, ensuring consistent representation across samples. The ALBERT model generates contextual embeddings by extracting the [CLS] token representation from the final hidden layer, producing a fixed-length embedding vector for each text instance. Embeddings are generated in batches to optimise memory utilisation and are stored for reuse to avoid repeated computation during model training. Each embedding vector is reshaped to form a sequence of length one, enabling compatibility with the LSTM layer used in subsequent processing.

The classification architecture consists of a single LSTM layer containing 128 units, followed by a fully connected dense layer with 64 units using ReLU activation and a final softmax output layer equal to the number of target classes. The model is trained using the Adam optimiser with a learning rate of 3e-4 and the sparse categorical cross-entropy loss function, making it suitable for multi-class classification with integer-encoded labels. Early stopping is applied based on validation loss to prevent overfitting, and a learning-rate reduction mechanism is employed to stabilise optimisation during training. The size of the batch will be 32 and the maximum number of epochs is sufficiently large and the network is trained until it reaches the early stopping mechanism. This combination of ALBERT features and LSTM layer acquires the capacity to grasp deep contextual semantics and allows the latter to acquire temporal and stylistic patterns that are inherent to human-written and text produced by the LLM.

# 6 Evaluation

This part offers the assessment and evaluation of the suggested methods through the comparison of the performances of various **machine learning, deep learning and hybrid models** on the test set. To determine the effectiveness of the models in the discrimination between human-written and AI-generated text, standard classification measures, such as **accuracy, precision, recall, and F1-score**, are used in order to have a clear and consistent comparison with one another.

## 6.1 Machine Learning Model Results

The best overall performance with **0.81** accuracy and a **0.82** F1 Score was obtained with the traditional machine learning baselines Random Forest, and it is backed by a comparatively higher macro precision (**0.83**), which means that the average number of false positives per class is smaller. Logistic Regression was very similar but a little less effective with an accuracy of **0.80** and a macro F1 of **0.81**, implying that it is also a competitive linear baseline but slightly worse at identifying complex patterns than the others. Linear SVM was the most accurate (**0.81**) and had the highest macro F1 (**0.82**) but with the best trade-off, with both precision and recall at approximately **0.82,** which means a similar level of classification accuracy between human and AI-generated categories.

**Table 2. Performance Comparison of Machine Learning, Deep Learning, and Hybrid Models**

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **Random Forest** | 0.81 | 0.83 | 0.81 | 0.82 |
| **Logistic Regression** | 0.80 | 0.82 | 0.81 | 0.81 |
| **SVM (Linear)** | 0.81 | 0.82 | 0.82 | 0.82 |
| **GRU** | 0.82 | 0.83 | 0.82 | 0.82 |
| **LSTM** | 0.82 | 0.82 | 0.82 | 0.81 |
| **ALBERT+LSTM** | 0.91 | 0.91 | 0.91 | 0.91 |

The results of the all the implemented models in terms of accuracy, precision, recall, and F1-scores are summarized in Table 2, which enables one to easily compare various modelling methods.

## 6.2 Deep Learning Models Results

In the case of the deep learning models, both GRU and LSTM provided a minor yet steady improvement over the classical baselines. GRU was found to have 0.82 accuracy, 0.83 precision, 0.82 recall, and 0.82 F1 Score with a slightly higher precision and overall balance. LSTM delivered a similar 0.82 accuracy and 0.82 precision/recall, with slightly lower F1

Score (0.81) hence the performance is similar to GRU but not as consistent across classes. On the whole, these findings indicate that recurrent networks can better learn sequential language patterns than the traditional models, but the improvement is not that significant at this point.

## 6.3 Superiority of the ALBERT+LSTM Architecture

The hybrid ALBERT+LSTM model attained an accuracy of 0.91, which is significantly higher than all baseline models. Its high contextual and sequential learning ability is confirmed by values of 0.91 of macro-average precision, recall, and F1-score.

The ALBERT+LSTM architecture was superior to any of the baseline machine learning and deep learning models because it can conjoin profound semantic insight and adequate sequence modelling. The transformer-based embedding mechanism of ALBERT is a rich semantic relationship, syntactic pattern, and contextually dependent word meaning embedding mechanism by leveraging self-attention operations, which allows the model to distinguish fine linguistic signals found in human and LLM-generated text. Its parameter-efficient architecture, such as factorised embeddings and cross-layer weight sharing, minimises computational costs without compromising the quality of representations. These contextual embeddings provide a more linguistic representation than the TF-IDF vectors or learned embeddings of the GRU and LSTM models, and lead to a higher separation of classes.

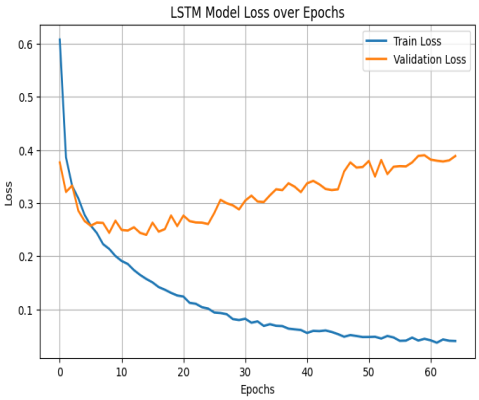


**Figure 7. LSTM Model Training and Validation Loss Over Epochs**

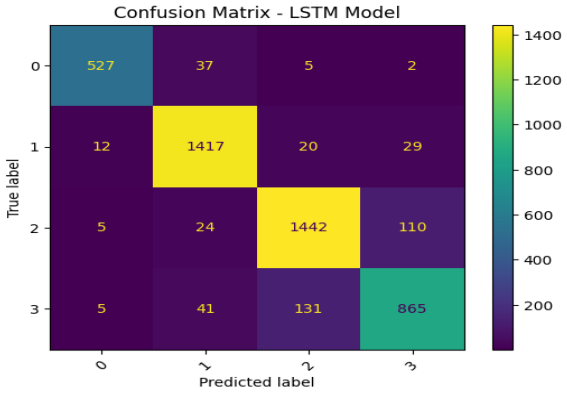


**Figure 8. Confusion Matrix for the LSTM Model**

Figure 7 illustrates that the training loss gradually diminishes as the model learns whereas the validation loss remains constant with minor variation. This trend shows that LSTM model is learning, but there is slight deviation, which reflects a weak overfitting during the course of time.

Figure 8 confusion matrix represents the accuracy of the LSTM model in classifying each of the classes in terms of the predicted labels and the actual labels. A high diagonal value means the performance is good and the lesser off-diagonal values represent the limited misclassification across the categories.

The next LSTM layer improves the work of the model by acquiring the ability to learn the temporal relations and the stylistic development of the embedding sequences. In as much as ALBERT offers powerful contextual features, the LSTM element allows the architecture to learn patterns of organization, coherence structure, and sequential variability that are typical of human-written text in comparison to text produced by a machine. This union enables the system to acquire global semantic information as well as local sequential patterns, which makes it have a better performance in all evaluation measures. Consequently, the hybrid method will yield the best accuracy, precision, recall, and F1-score, which showed the effectiveness of the hybrid method in working with short, noisy, and linguistically diverse social media text.

## 6.4 Comparison of All Models

The analysis shows that the TF-IDF + machine learning models (Random Forest, Logistic Regression, Linear SVM) generate strong baselines with a combination of metrics that tend to cluster around 0.80-0.82, but largely because of the superficial nature of the signals that word-frequency provides, do not reflect on more profound contextual and semantic differences between human and LLM text. There is a small improvement in the recurrent deep learning models (GRU, LSTM), as these models learn word-order dependence and sequential dependence, but their representations fail to reveal the rich contextual meaning needed to do fine-grained discrimination. Comparatively, the hybrid ALBERT+LSTM model, which introduces a vital improvement of the 0.91 of the measures of accuracy, precision, recall, and F1 since ALBERT provides context-sensitive transformer embeddings, gathering semantic nuances, sequence-level flow and stylistic rhythm, provides a substantial increase in both the accuracy and the precision, recall, and F1.

# 7 Conclusion and Future Work

## 7.1 Conclusion

This research paper entailed a thorough investigation into the task of differentiating between human-written text and the one produced by large language models on the basis of machine and deep learning and hybrid architectures. The systematic data preparation, feature engineering, model development, and performance evaluation were achieved with the help of a structured approach and the CRISP-DM framework. The comparative study proved that conventional machine learning designs based on TF–IDF vectorisation provided a strong set of baselines but were not rich enough in semantic understanding to identify the contextual nuances of the modern AI-generated text. Recurrent deep learning networks such as GRU and LSTM network demonstrated superior performance through the modelling of sequential dependencies; but had limited capacity when modelling highly contextual patterns generated by more advanced LLMs.

The hybrid ALBERT+LSTM model was the most successful in all the metrics and it showed a considerable improvement in accuracy, precision, recall, and F1-score. Such an

enhancement can be explained by the contextual embedding features of ALBERT that retrieve semantic relations and syntactic structure, and the capability of the LSTM layer to learn temporal and stylistic patterns. The model demonstrates a high level of generalisation with numerous text types, including GPT-3.5, Claude-v1, LLaMA-30B, and human-generated texts, which suggests that it can be used in the identification of text authenticity in real-world situations, especially in social media, where the linguistic variations are large. The results support the topicality of hybrid transformer-recurrent architectures in solving new problems of digital content verification.

## 7.2 Future Work

Further research could be developed in the future by using more and more varied data sets such as multilingual corpora and text sources specific to domain to further evaluate generalisability across language and cultural backgrounds. Further work on other lightweight transformer architecture models, like Distil BERT or ALBERT-xxlarge, can be insightful regarding efficiency-performance trade-offs. The incorporation of explainability methods would also improve transparency by finding linguistic indicators affecting model decisions, which can be more widely applied in the context of digital forensics and content moderation. Additionally, the implementation of the hybrid architecture into real-time detection pipelines would allow to test it under realistic conditions on operational limitations and would provide the chances to study the scalability, latency optimisation, and constant adaptation of the model in changing online conditions.

# References


1. Ahammad, S., Sinthia, S. A., Ahmed, M., Hossain, M., Asif, N. A. A., & Ikram, N. A. (2024, April). *Deep learning-based sentiment analysis of user generated reviews of various AI powered mobile applications*. In *2024 International Conference on Inventive Computation Technologies (ICICT)* (pp. 505–512). IEEE.
2. Alsubari, S. N., Deshmukh, S. N., Alqarni, A. A., Alsharif, N., Aldhyani, T. H., Alsaade, F. W., & Khalaf, O. I. (2022). Data analytics for the identification of fake reviews using supervised learning. *Computers, Materials & Continua, 70*(2), 3189–3204.
3. Bhuvaneshwari, P., Rao, A. N., & Robinson, Y. H. (2021). Spam review detection using self-attention-based CNN and bi-directional LSTM. *Multimedia Tools and Applications, 80*(12), 18107–18124.
4. Blake, J., Miah, A. S. M., Kredens, K., & Shin, J. (2025). Detection of AI-generated texts: A Bi-LSTM and attention-based approach. *IEEE Access*.
5. Chandana, I., Reshma, O. M., Sree, N. G., Reddy, B. J., & Shareefunnisa, S. (2024, July). *Detecting AI generated text*. In *2024 2nd World Conference on Communication & Computing (WCONF)* (pp. 1–6). IEEE.
6. Elmogy, A. M., Tariq, U., Ammar, M., & Ibrahim, A. (2021). Fake reviews detection using supervised machine learning. *International Journal of Advanced Computer Science and Applications, 12*(1).

7. Fagni, T., Falchi, F., Gambini, M., Martella, A., & Tesconi, M. (2021). TweepFake: About detecting deepfake tweets. *PLOS ONE, 16*(5), e0251415.
8. Ghatora, P. S., Hosseini, S. E., Pervez, S., Iqbal, M. J., & Shaukat, N. (2024). Sentiment analysis of product reviews using machine learning and pre-trained LLM. *Big Data and Cognitive Computing, 8*(12), 199.
9. Hireche, A., Al-Dabet, S., Mediani, M., & Belkacem, A. N. (2025, April). *Detecting AI-generated text: A Bi-GRU with linguistic features approach*. In *2025 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1–7). IEEE.
10. Kayabas, A., Topcu, A. E., Alzoubi, Y. I., & Yıldız, M. (2025). A deep learning approach to classify AI-generated and human-written texts. *Applied Sciences, 15*(10), 5541.
11. Mitrović, S., Andreoletti, D., & Ayoub, O. (2023). ChatGPT or human? Detect and explain: Explaining decisions of machine learning models for detecting short ChatGPT-generated text. *arXiv preprint arXiv:2301.13852*.
12. Rosca, C. M., Stancu, A., & Iovanovici, E. M. (2025). The new paradigm of deepfake detection at the text level. *Applied Sciences, 15*(5), 2560.
13. Sadiq, S., Aljrees, T., & Ullah, S. (2023). Deepfake detection on social media: Leveraging deep learning and FastText embeddings for identifying machine-generated tweets. *IEEE Access, 11*, 95008–95021.
14. Wani, M. A., ElAffendi, M., & Shakil, K. A. (2024). AI-generated spam review detection framework with deep learning algorithms and natural language processing. *Computers, 13*(10).
15. Yang, J. L. (2025). Classification of artificial intelligence-generated product reviews on Amazon. *Engineering Proceedings, 92*(1), 17.