ASEAN
Journal of Scientific and Technological Reports
Online ISSN:2773-8752

Research article

Bayesian Meta-Ensemble of Stacked Long Short-Term Memory Convolutional Neural Networks for Detection of AI-Generated Text under Uncertainty

Nirawit Kanthachai[1], and Prompong Sugunnasil[2*]

[1] Faculty of Engineering, Chiang Mai University, Chiang Mai, 50200, Thailand
[2] College of Art, Media and Technology, Chiang Mai University, Chiang Mai, 50200, Thailand
* Correspondence: prompong.sugunnasil@cmu.ac.th

Citation:
Kanthachai, N.; Sugunnasil, P. Bayesian neta-ensemble of stacked long short-term memory convolutional neural networks for detection of ai-generated text under uncertainty. *ASEAN J. Sci. Tech. Report.* **2026**, *29*(5), e261528.. https://doi.org/10.55164/ajstr.v29i5.261582

Article history:
Received: September 28, 2025
Revised: March 24, 2026
Accepted: April 11, 2026
Available online: April 30, 2026

Publisher's Note:
This article is published and distributed under the terms of the Thaksin University.

Abstract: The rapid misuse of AI-generated text can lead to a serious risk to information integrity, yet most existing detectors rely on deterministic, single-model architectures that lack uncertainty quantification and are computationally demanding. This paper introduces a Bayesian Meta-Ensemble of stacked LSTM-CNNs for detecting AI-generated text, a lightweight probabilistic framework that addresses these limitations by combining ensemble stacking with variational Bayesian inference. Three main novel contributions: (i) a two-tier stacked ensemble trained on resampled subsets feeding into a Bayesian Neural Network (BNN) meta-learner, enabling uncertainty-aware classification, (ii) a statistical analysis of Monte Carlo sampling behavior using Kolmogorov-Smirnov and Mann-Whitney tests to validate predictive stability, and (iii) the integration of parametric and non-parametric hypothesis testing directly into the prediction pipeline to enhance decision reliability. On the MAGE dataset, we benchmarked our model against several state-of-the-art detectors, including FastText, GLTR, Longformer, and DetectGPT. Our model achieves 90.84% human recall, 89.59% average recall, and 0.96 AUROC, outperforming FastText, GLTR, and DetectGPT, and closely matching Longformer while requiring 7.5 times fewer parameters (18.77M vs. 142M). We further investigated predictive stability through normality testing and examined the effect of the Monte Carlo (MC) sample size, confirming that larger samples led to more consistent model outputs. Additionally, we demonstrated the use of statistical inference to enhance decision-making. The results demonstrate that a compact Bayesian ensemble can deliver competitive detection accuracy while accounting for uncertainty estimates, making it suitable for resource-constrained and high-stakes environments.

The code is available at https://github.com/Filimize/CNN_LSTMs_BNN.

Keywords: Bayesian model; uncertainty estimation; variational inference; model interpretability; ai-generated text detection

1. Introduction

In the age of advanced technology, AI is used for various purposes, including text generation. The introduction of Large Language Models (LLMs) [1] has attracted many people to use them for beneficial applications such as queries and question answering. However, it can also cause mayhem in both the

digital realm and broader society [2] since LLMs are capable of generating fake news, spreading misinformation, and misleading advertising [3]. Consequently, developing an effective tool to distinguish AI-generated text from human-written text has become a top priority. One approach to mitigating the harmful effects of LLMs is to develop detector tools capable of distinguishing between AI-generated text and human writing, which can reduce the potential for misuse [4]. Prior studies [4, 5] suggest that an ideal detector should be accurate, with a good balance between false positives and false negatives depending on the tasks, require as few samples as possible for the training process, generalize with different models that attackers use, produce interpretable decisions, and robust to adversarial inputs[7]. There are several AI-generated text-detection models, each with its own advantages. For example, Bidirectional encoder representations from transformers (BERT) [8] uses a bidirectional approach and a transformer architecture, which helps capture long-range word relationships. DistilBERT [9] offers a smaller, faster, but slightly lower-accuracy model than BERT. RoBERTa [10] improves upon BERT by training with longer sequences and more training data, making it more accurate than BERT. Despite strong performance, these models produce deterministic outputs and lack mechanisms for quantifying prediction uncertainty.

Uncertainty quantification is critical in high-stakes classification tasks, where overconfidence or underconfidence can lead to serious harm [11]. Bayesian Deep Learning introduces probabilistic principles into deep learning models, producing uncertainty estimates through posterior inference over model weights [12], but it struggles to estimate the posterior distribution of network weights, leading to poor predictive uncertainty and calibration. There are several algorithms to estimate the distribution, including Variational Inference (VI) and Markov Chain Monte Carlo (MCMC). Although MCMC is considered the gold standard for distribution estimation, VI is significantly more computationally efficient. Therefore, VI methods are frequently used when working with large datasets. Despite progress in Bayesian deep learning and ensemble methods individually, their combination as a meta-learning architecture for AI-generated text detection with uncertainty quantification and statistical inference for decision support remains underexplored. Existing ensemble-based text detectors typically stack deterministic classifiers without propagating predictive uncertainty, limiting their reliability in real-world deployment.

To address these gaps, we present the Bayesian Meta-Ensemble of Stacked LSTM-CNNs for AI-generated text detection. The proposed model introduces three novel contributions: (i) a two-tier stacked ensemble in which lightweight LSTM-CNN base models, trained on resampled subsets to handle class imbalance, feed probabilistic outputs into a Bayesian Neural Network (BNN) meta-learner, enabling uncertainty-aware classification, (ii) a statistical analysis of Monte Carlo sampling behavior using KS and Mann-Whitney tests to validate predictive stability, and (iii) integration of parametric and non-parametric hypothesis testing directly into the prediction pipeline to support decision-making beyond a fixed threshold. The Machine-Generated Text Detection (MAGE) dataset includes human-written texts from different sources and machine-generated texts produced by 27 large language models (LLMs) to test the benchmark [13]. Variational Inference is employed to approximate the posterior distribution over network weights, and the development follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework.

2. Materials and Methods

2.1 Planning

2.1.1 Study related research, including methods for improving Bayesian Neural Network (BNN) performance, LSTM-CNN for text classification, enhancements to word embeddings, and societal issues arising from AI-generated text.

2.1.2 Design procedures for developing the Bayesian Meta-Ensemble of Stacked LSTM-CNNs model using the Cross-Industry Standard Process for Data Mining (CRISP-DM).

2.1.3 Develop the classification model following the CRISP-DM framework.

2.1.4 Measure the performance of the model using appropriate evaluation metrics.

2.1.5 Summarize and discuss the results in the context of model effectiveness and broader implications.

2.1.6 Develop a Bayesian Meta-Ensemble of Stacked LSTM-CNNs model for AI-generated text detection using the MAGE dataset.

2.1.7 Measure the performance of the Bayesian Meta-Ensemble of Stacked LSTM-CNNs model for AI-generated text detection, compared with the well-known text classification models: FastText, GLTR, Longformer, and DetectGPT.
Prediction performance: Recall of AI-generated text, recall of human-written text, average recall, and area under the ROC curve (AUC-ROC).
Computational performance: Model size.

2.1.8 Compare the Bayesian Meta-Ensemble of Stacked LSTM-CNNs model's performance against FastText, GLTR, Longformer, and DetectGPT [13].

2.1.9 Experiment with Monte Carlo sample size and statistical inferences.

2.1.10 Conclude and discuss the results.

2.2 Scope

This paper uses the Machine-Generated Text Detection (MAGE) dataset, which includes human-written texts from different sources and machine-generated texts produced by 27 large language models (LLMs). The dataset contains a collection of human- and AI-generated text from various sources. The dataset is organized into three subsets.

Training set: 225,753 records of AI-generated text and 93,318 records of human text.

Validation set: 27,993 records of AI-generated text and 28,799 records of human text.

Testing set: 30,478 records of AI-generated text and 30,265 records of human text, including some examples from sources outside the training and validation sets.

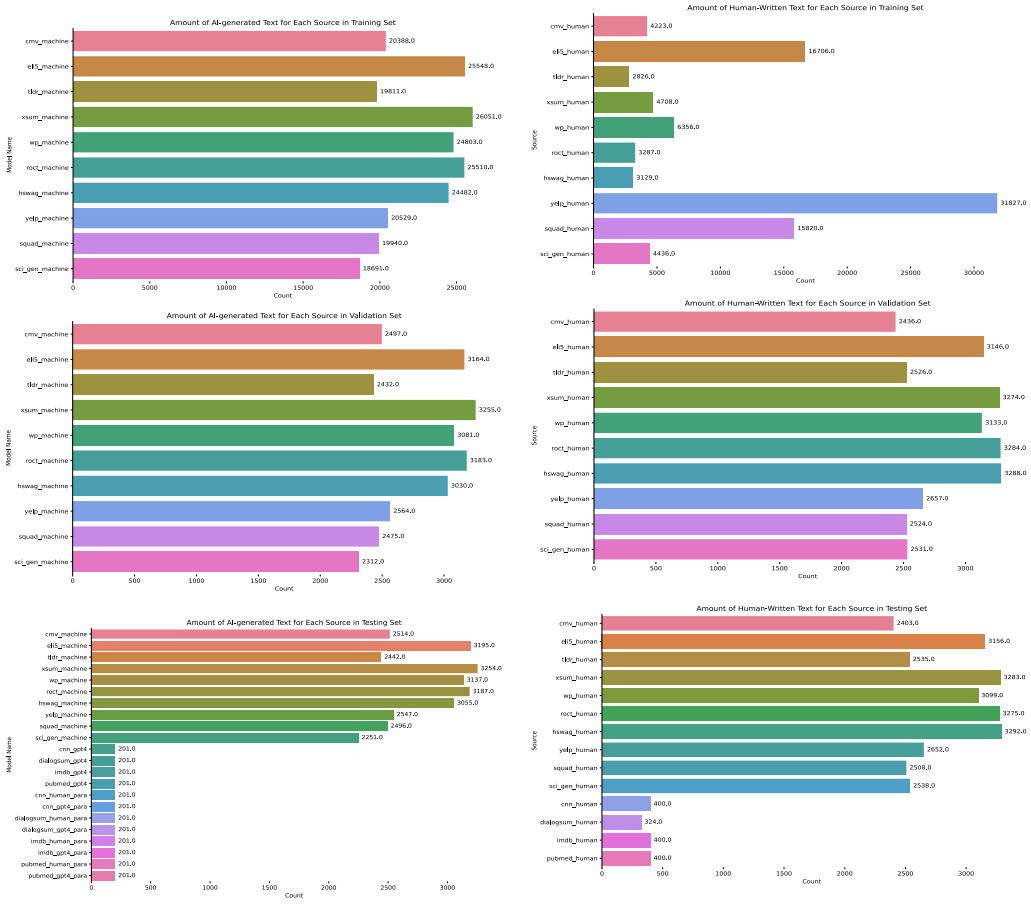


Figure 1. Amount of data categorized by the sources of the data in each dataset.

2.3 Literature Review: Data Science Literature

2.3.1 Long Short-Term Memory Model (LSTM)

Even though Recurrent Neural Networks (RNNs) have been widely used in text processing due to their ability to handle variable-length sequences, they are prone to the vanishing gradient problem. This issue arises because RNNs are equivalent to Multi-Layer Perceptrons (MLPs) with a large amount of historical information, which makes RNNs struggle with long sequences. The accumulation of historical information can lead to vanishing gradients, where the gradients become exceedingly small. As a result, the neural network's nodes rarely update, leading to slow convergence and difficulty in learning long-term dependencies. On the other hand, the Long Short-Term Memory Model (LSTM), an improved RNN-based network, can retain historical information over long sequences by adding memory cells and three control gates, mitigating the loss of historical information and the vanishing gradient problem caused by excessive RNN layers.

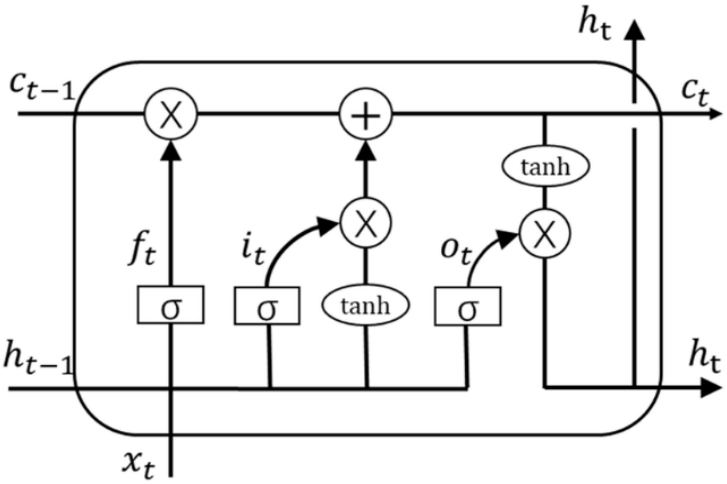


Figure 2. The description of LSTM node.

The LSTM structure is shown in Figure 2. The memory cell is added for storing the historical information, and three main gates control the update, deletion, and output of the information: the input gate determines how the incoming vectors alter the state of the memory cell, the output gate allows the memory cell to affect the outputs, and the forget gate enables the memory cell to remember or discard previous information [14]. Xiao et al. [15] presented a paper on patent text classification, showing that a pre-trained Word2Vec LSTM model effectively mitigates the dimensionality disaster typically encountered with traditional methods. In this experiment, 50,000 patent texts were used for training and testing, yielding a test accuracy of 93.48%, which outperformed other models such as CNNs and k-nearest neighbors. Xie et al. [16] proposed "Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification," introducing a Self-Attention-Based BiLSTM model to improve sentiment polarity classification by incorporating aspect-term information into the learning process. Their proposed Bidirectional LSTM (BiLSTM) leverages both past and future contexts, allowing for more comprehensive semantic feature extraction. Additionally, the integration of a self-attention mechanism enhances the model's ability to focus on crucial aspect-term information while reducing computational complexity. In comparative experiments, the Self-Attention-Based BiLSTM model achieved higher accuracy than standard LSTM and BiLSTM models in sentiment polarity classification. However, the study identified limitations, particularly in distinguishing between Negative and Neutral sentiment categories due to unclear boundary definitions. Wadud et al. [17] proposed using an LSTM with the AdaBoost algorithm for text classification on a Bengali dataset. This method outperformed Bi-LSTM CNN and CNN-LSTM models, achieving a precision of 89.48%, a recall of 93.38%, and an F1-score of 91.39%.

2.3.2 Embedding

Embedding is the process of converting text into vector representations that capture the semantic meaning. In text classification, models such as LSTMs typically use embeddings, for instance, Word2Vec. Anggrainingsih et al. [18] proposed an MLP model using BERT embeddings as input for detecting text misinformation, compared with SOTA models. Their experimental results show that their proposed model outperforms SOTA models and performs better when paired with RoBERTa embeddings than with BERT or DistilBERT.

2.3.3 Convolutional Neural Networks (CNNs)

CNNs are designed to address the limitations of traditional neural networks, which tend to struggle with the high computational demands of computer vision tasks. Common machine learning benchmark datasets, such as MNIST, are databases of handwritten digits collected in black-and-white images of size 28x28 pixels, which are manageable for traditional neural networks. For these images, the input layer contains 784 nodes, which is manageable for traditional neural networks. However, for an RGB image of size 128x128 pixels, the input layer would contain 49152 nodes. To capture the patterns in these images, several hidden layers and nodes must be added, increasing computational intensity and the risk of overfitting. This is why Convolutional Neural Networks (CNNs) are preferred over traditional neural networks for computer vision tasks. CNNs are specifically designed to handle spatial hierarchies in images, reducing parameter count and computational complexity while capturing important features. CNNs are comprised of three types of layers: convolutional layers, pooling layers, and fully connected layers. These layers are stacked to form a CNN architecture. A CNN architecture has been formed. The key areas of the CNN can be broken down as follows:
Input Layer: This layer stores the image's pixel values.
Convolutional Layer: Determines the output of neurons connected to local regions of the input by calculating the scalar product between their weights and the connected input volume.
Pooling Layer: Performs downsampling along the spatial dimensions of the input, reducing the number of parameters in the activation.
Fully Connected Layers: Perform the same functions as in standard neural networks, producing class scores from the activations for classification purposes [19].

Kim [20] proposed the use of CNNs for sentence classification in his paper "Convolutional Neural Networks for Sentence Classification". Through a series of experiments, the study demonstrated that using Word2Vec as an embedding, even a simple CNN with a single convolutional layer, could achieve impressive performance. This led to the conclusion that well-established word vectors play a crucial role in deep learning for Natural Language Processing (NLP). Alshubaily [21] proposed incorporating an attention mechanism to boost performance and introduced WordRank for vocabulary selection, thereby reducing the number of embedding parameters and speeding up training with minimal accuracy loss. Implementing these ideas improved TextCNN accuracy on the 20News dataset from 94.79% to 96.88%, while reducing embedding layer parameters from 7.9M to 1.5M with only a 1.2% decrease in accuracy. In 2022, Zhao et al. [22] further built on Yoon Kim's ideas by proposing the WTL-CNN method, which combines Word2Vec, a topic-based TF-IDF algorithm, and an improved convolutional neural network. Initially, Word2Vec converts text data into word vectors. An improved TF-IDF algorithm incorporating the LDA topic generation model is then used to weight these vectors and enhance their semantic topic information. Furthermore, the location information of important features in news texts is converted into weights and integrated into the convolutional neural network's pooling layer, thereby boosting classification accuracy. WTL-CNN was evaluated against seven models using THUCNews and SogouCS datasets in a TensorFlow environment. Experimental results revealed that the model achieved precision, recall, and F1 scores of 95.76%, 93.43%, and 94.98%, respectively, on THUCNews, and 94.61%, 93.43%, and 94.01%, respectively, on SogouCS.

2.3.4 LSTM-CNN

Neural Networks have achieved outstanding results in Natural Language Processing (NLP). Using word embeddings to represent text can yield semantic representations of words and phrases, not only capturing the characteristics of word relationships in sentences but also avoiding the problems of high

dimensionality and data sparsity [23]. One suitable model for text classification is the LSTM-CNN, a hybrid model combining LSTM and CNN. This model is mainly composed of three parts: word embedding for word representation as numerical values, an LSTM-CNN model, and an output layer that uses either the activation function "softmax" for multiple-class classification or "sigmoid" for binary classification.

C. Li et al. [24] developed the Bi-LSTM-CNN and tested the model's performance through the THUCNews dataset, which contains a training set of 50000 corpora, a validation set of 5000 corpora, and a testing set of 10000 corpora. The authors compared the Bi-LSTM CNN with other models. The results are shown in Figure 2. The results suggest that the Bi-LSTM-CNN model effectively utilizes the loop structure to obtain contextual information. It constructs the left and right contexts for each word using a Convolutional Neural Network (CNN), thereby achieving a more accurate representation of the text's semantics. However, the author suggests that using attention mechanisms could further improve the model's performance.

Table 1. The comparison experiment results from C. Li et al. *[24, p. 201]* "News Text Classification Based on Improved Bi-LSTM-CNN".

	The comparison experiment results		
Model	**Accuracy (%)**	**Loss**	**F1 Score**
TF-IDF	90.27	0.31	0.86
SVM	93.49	0.25	0.92
LSTM	94.26	0.21	0.94
CNN	95.61	0.15	0.96
Bi-LSTM-CNN	96.46	0.11	0.99

Jang et al. [25] proposed the "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec, CNN, and Attention Mechanism." This Bi-LSTM-CNN hybrid model was designed for sentiment classification using the Internet Movie Database (IMDB) review dataset. The approach uses a CNN for feature extraction, reducing the input feature space, while a Bi-LSTM captures contextual dependencies within the extracted features. Additionally, the authors incorporated an attention mechanism. Experimental results demonstrate that the Bi-LSTM-CNN hybrid model significantly outperforms baseline models in both accuracy and F1-score. The experiment was conducted on a fixed dataset size and yielded an accuracy of 0.9141 for the proposed model, outperforming CNN (0.8874), LSTM (0.8940), Multi-Layer Perceptron (MLP) (0.7129), and an existing hybrid model (0.8906). Similarly, the F1-score of the proposed model (0.9018) surpassed those of CNN (0.8875), LSTM (0.8816), MLP (0.7708), and the existing hybrid model (0.8887).

2.3.5 Bayesian Neural Network (BNN)

BNN is a probabilistic model based on Bayes' Theorem. BNN combines prior "expert" information and the likelihood of the data to produce the posterior distribution. The BNNs have several benefits; one is that they can handle Aleatoric and Epistemic uncertainty. Aleatoric uncertainty is inherent in the data itself, such as noise or anomalies, while Epistemic uncertainty is derived from the model structure and hyperparameters [26]. However, the learning process of BNNs is based on inferring the posterior distribution by sampling from it using Bayes' theorem; therefore, it requires combining the prior and likelihood. Obtaining posterior samples is difficult and computationally intensive, especially given the high dimensionality of the sampling space. In some situations, the prior and the likelihood may have different distributions; therefore, BNNs need to use sampling algorithms such as Markov Chain Monte Carlo (MCMC), Variational Inference (VI), and MC-Dropout to estimate the posterior, which are considered approximation algorithms. In a Bayesian Neural Network, performing Bayesian inference requires the posterior distribution of the network weights. Given the prior probability of network weights $P(w \mid \alpha)$ which depends on parameters α , then the posterior can be written as $Pr(w \mid D, \alpha)$ However, for most neural networks, it is difficult to obtain the posterior probability of weights; therefore, Variational inference is used to address this problem by approximating

$Pr(w \mid D, \alpha)$ with a more tractable distribution $Q(w \mid \beta)$ [27]. This approximation is fitted by minimizing the variational free energy F with respect to β where

$$F = -\left\langle \ln\left[\frac{\Pr(D \mid w)\, P(w \mid \alpha)}{Q(w \mid \beta)}\right]\right\rangle_{w\sim Q(\beta)}$$

Graves [27] tested combinations of posteriors and priors on a hierarchical multidimensional recurrent neural network for phoneme recognition on the TIMIT speech corpus. A pruning heuristic was also applied to the trained model. The results showed that using Gaussian distributions as both prior and posterior gave the best performance in terms of minimizing the error, and they also found that the pruning methods enhance the model's performance by significantly reducing error when the threshold for pruning is strong enough (threshold ≥ 0.05)

2.3.6 AI detection using ensemble models

Ensemble learning improves generalization by reducing the risk of overfitting to a particular linguistic pattern or model-specific bias. By aggregating diverse decision boundaries, it achieves a more robust and reliable classification than any individual model could provide alone. In the context of AI-generated text detection, where the linguistic overlap between human-written and machine-generated text is continually narrowing, ensemble architectures have consistently improved performance over single-model baselines across multiple benchmark datasets. Abburi et al. [28] proposed an ensemble framework for AI-generated text detection that concatenates output probability vectors from five pre-trained transformer models: DeBERTa-large, XLM-RoBERTa, RoBERTa-large, RoBERTa-base OpenAI detector, and XLM-RoBERTa NLI, and feeds the combined representation into a voting classifier to produce the final prediction. Aggarwal et al. [29] similarly proposed an ensemble framework combining BERT and DeBERTa with a dynamic weight adjustment strategy, where each constituent model's contribution is weighted by its validation accuracy, and a cosine learning rate scheduler is used to improve convergence stability. Although both ensemble models achieved strong detection accuracy, they share common limitations. All constituent models and voting classifiers produce deterministic outputs, offering no mechanism for quantifying prediction uncertainty; a significant drawback in high-stakes applications such as misinformation detection and academic integrity verification. Furthermore, both frameworks rely on large transformer-based models (DeBERTa-large with approximately 400M parameters and RoBERTa-large with 356M parameters), making them computationally expensive and impractical for resource-constrained environments. Additionally, neither study explicitly addresses class imbalance in the training data. The proposed framework addresses these limitations by (i) replacing the deterministic voting classifier with a Bayesian Neural Network meta-learner that produces calibrated probabilistic outputs and supports statistical hypothesis testing, (ii) using lightweight LSTM-CNN base models with fewer than 6 million parameters each, achieving competitive performance at a fraction of the computational cost, and (iii) applying a data resampling strategy across multiple balanced training subsets to mitigate class imbalance effects across all base models.

2.4 Literature Review: Statistics Literature

2.4.1 Kolmogorov-Smirnov Test

To test the hypothesis about a population distribution, we sometimes investigate the agreement between the sample distribution and the theoretical distribution, a procedure known as a "goodness of fit test". Some tests have been developed in which the sample distribution depends on the form or the values of certain parameters in the population distribution. In contrast, test statistics whose sampling distribution does not depend on either the explicit form or the value of certain parameters are called nonparametric or distribution-free tests [30]. For testing whether two samples come from the same distribution, the Kolmogorov-Smirnov goodness-of-fit test, a nonparametric test, is used. Suppose that the first sample has size m with a cumulative distribution function (CDF) denoted by $F(x)$ and the second sample has size n with a CDF denoted by $G(x)$. Define

$$D_{m,n} = \max_x |F(x) - G(x)|$$

The null hypothesis is that both samples come from the same population distribution. We reject the null hypothesis if $D_{m,n} > D_{m,n,\alpha}$ where $D_{m,n,\alpha}$ is the critical value. For m and n sufficiently large

$$D_{m,n} = c(\alpha)\sqrt{\frac{m+n}{mn}}$$

Where $c(\alpha)$ is the inverse of the Kolmogorov distribution at α [31].

2.4.2 Mann-Whitney U test

The Mann-Whitney U test is a nonparametric statistical test in which the null hypothesis states that random samples from two populations have the same distribution. The Mann-Whitney U test is often misunderstood as an alternative to the t-test for testing median equality, but it can be used only to test median equality if it is assumed that the two populations have identical shapes and differ only by a shift [32]. Feltovich [33] compared the Mann-Whitney test and the robust rank-order test across several simulation examples. They found that when the distributions have the same dispersion, and are identical but not symmetric, the test can yield valid results even though it is not theoretically ideal. They also noted that the Type I error (the error of rejecting the null hypothesis when it should not) was significantly reduced with larger sample sizes (60 per group). Yue and Wang [34] investigated the power of the Mann-Whitney test in detecting shifts in the median or mean of time series data. The study found that the power of the test increases with the magnitude of the shift, sample size, and significance level, and decreases with higher data variation.

2.4.3 Student's t-test

The statistical test is used to determine whether the difference between two groups of data is statistically significant. The test follows a Student's t-distribution under the null hypothesis. It is normally applied when the data is normally distributed. The null hypothesis states that the means are statistically equal. There are three types of t-test: one-sample t-test, independent samples t-test, and paired samples t-test. The one-sample t-test is used to determine whether the mean of a sample is statistically equal to the expected value. It is typically used when the sample size is less than 30. If the sample size is 30 or greater, the z-test is preferred. However, the z-test requires the population standard deviation to be known. If the population standard deviation is unknown, the t-test can be used instead [35]. The t-test assumes the data are normally distributed, but most real-world data do not follow a normal distribution. Several papers suggest that the t-test can still be used, provided the sample size is sufficiently large, and the data are not normally distributed [36]. Ratcliffe [37] investigated the effects of sample size on the power of the t-test; the results suggest that as the sample size increases, the influence of non-normality decreases significantly. He noted that "extreme non-normality can as much as double the value of t at the 2.5% (one tail) probability level for small samples, but increasing the sample sizes to 80, 50, 30, and 15 will, for practical purposes, remove the effect of extreme skewness, moderate skewness, extreme flatness, and moderate flatness, respectively." Sawilowsky and Blair [38] tested the t-test across various simulation examples; the results show that the t-test is reasonably robust to type I error under most conditions. However, small or unbalanced samples and extreme non-normal distributions can lead to conservatism. The power, also called Type II error (the error in which the test does not reject the null hypothesis when it should), remains fairly stable across different distribution shapes and even under some violations of assumptions, especially with large sample sizes.

2.4.4 Shapiro-Wilk test

The Shapiro-Wilk test is used to test the null hypothesis that a sample comes from a normally distributed population. This test can detect departures from normality due to skewness, kurtosis, or both [39]. Mohd Razali and Yap [40] investigated the power of different normality tests: the Kolmogorov-Smirnov test, Lilliefors test, Anderson-Darling test, and Shapiro-Wilk test, using Monte Carlo procedures. The results showed that the Shapiro-Wilk test outperformed other tests, especially with large samples. Although the

Shapiro-Wilk test's power is relatively low for small sample sizes, it remains comparable to other tests, whereas the Kolmogorov-Smirnov test has the least power.

2.5 Literature Review: Domain Knowledge Literature

2.5.1 Potential Harm of AI-generated Text

The introduction of Large Language Models (LLMs) has led to AI-generated writing being used across fields such as journalism and marketing. In contrast, AI writing has many advantages. On the other hand, AI writing can be misused for manipulating information or propaganda, leading to social unrest and instability in politics. For instance, in 2019, OpenAI decided not to release the full version of GPT-2 due to concerns about potential misuse [41]. In 2022, "AI's Jurassic Park Moment – Communications of the ACM" [42] discussed that some LLMs are inherently unreliable, leading to AI hallucinations, The article raised concerns about the reliability and trustworthiness of LLMs, and provided examples of the harm caused by AI-text generated: for example, people used OpenAI to flood the Stack Overflow website with AI responses that had a high rate of incorrectness [43], prompting the site to ban ChatGPT-generated submission. Gary Marcus [41] proposed the "four paths" to mitigate the threats posed by AI-generated text. Firstly, every social media company and search engine should ban the generation of misleading content. Secondly, every country must reconsider its policies regarding misinformation. Thirdly, a stronger validation system should be implemented, with all websites applying anonymous, bot-resistant authentication. Fourth, a new kind of AI should be built to counter these threats, such as models that can verify truthfulness.

Dergaa et al. [44] gathered a literature review of relevant scholarly articles published in peer-reviewed journals indexed in Scopus as quartile 1, using keywords such as "ChatGPT," "AI-generated text," "academic writing," and "natural language processing". The author used a quasi-qualitative approach, involving reading and evaluating sources and identifying relevant data to support the research questions. The study found that ChatGPT and other NLP technologies can enhance the quality of academic writing and research. However, the research found that ChatGPT produced responses containing factual errors and misrepresentations, which could be attributed to the absence of relevant articles in its training set and/or its inability to differentiate between credible and less credible sources. The paper also mentions a sting operation conducted by John Bohannon [45]. Over a period of 10 months, Bohannon submitted a fake scientific paper to 304 open-access journals. His fake paper contained a fatal flaw and fabricated content. Shockingly, the fake paper was accepted by 157 journals. This raises significant concerns about the quality control of open-access journals, especially today, when AI-generated text models are easy to access.

3. Results and Discussion

3.1 Base Models Results

The training data is class-imbalanced: 225753 records of AI-generated text and 93318 records of human text. To address this, the training data was split into three balanced subsets: each contains 75251 records of AI-generated text (drawn from distinct partitions) and the full 93318 records of human text. The original imbalanced dataset was retained as a fourth training set for comparison.

3.2 Model comparison

After training the base models, a Bayesian neural network was constructed as the meta learner using Variational Inference (VI). The meta-learner architecture is a 4-(12-10)-1 fully connected network. The model was evaluated on human-written texts from all domains and AI-generated texts from all LLM sets in the MAGE dataset. Performance was compared against FastText, GLTR, DetectGPT, and the Longformer model, whose metrics were originally reported by Y. Li et al. [13]

Table 2. The results of each base model with different balanced datasets

Training Dataset	Model Name	Accuracy		Loss	
		Training	Validation	Training	Validation
The 1st set	**LSTM-CNN**	0.9378	0.8904	0.1506	0.2638
	BiLSTM-CNN	0.9147	0.8682	0.2022	0.2994
	Multi-Head LSTM-CNN	0.9397	0.8817	0.1493	0.2782
The 2nd set	**LSTM-CNN**	0.9445	0.8874	0.1372	0.2697
	BiLSTM-CNN	0.9456	0.8805	0.1328	0.2769
	Multi-Head LSTM-CNN	0.9460	0.8878	0.1326	0.2769
The 3rd set	**LSTM-CNN**	0.9475	0.8883	0.1304	0.2636
	BiLSTM-CNN	0.9107	0.8619	0.2126	0.3094
	Multi-Head LSTM-CNN	0.9108	0.8645	0.2130	0.3095
The 4th set	**LSTM-CNN**	0.9265	0.8943	0.2404	0.2558
	BiLSTM-CNN	0.9254	0.8756	0.2495	0.2856
	Multi-Head LSTM-CNN	0.9303	0.8944	0.2327	0.2579

Across all datasets, Table 2 shows that the best-performing model is regular LSTM-CNN + ALBERT embeddings. All developed models contain fewer than 6 million parameters; this constraint likely explains why the simpler LSTM-CNN outperforms the more complex architectures, as neither BiLSTM-CNN nor Multi-head Attention LSTM-CNN has sufficient capacity to exploit their additional components at this parameter scale fully.

Table 3. The results of our model compared with the baselines

Model	Human Recall	AI Recall	Avg Recall	AUROC
FastText	86.34%	71.26%	78.80%	0.83
GLTR	12.42%	98.42%	55.42%	0.74
Longformer	82.80%	98.27%	90.53%	0.99
DetectGPT	86.92%	34.05%	60.48%	0.57
LSTM-CNN+BNN (MC= 1000)	90.84%	88.34%	89.59%	0.96

The proposed LSTM-CNN+BNN model achieves the highest human recall among all compared detectors. Its average recall is 0.94% below Longformer, and its AUROC is 0.03 lower, while using only 18.77M parameters, approximately 7.5 times fewer than Longformer (142M). It should be noted that RoBERTa-based detectors and dedicated GPT-detector models, which were not included in the original MAGE benchmark comparison used here, have been reported to achieve strong performance in the literature but similarly depend on large-scale transformer architectures that may be impractical in resource-limited settings. The proposed model represents a favorable trade-off between accuracy, uncertainty quantification, and computational efficiency.

3.3 Monte-Carlo sampling result

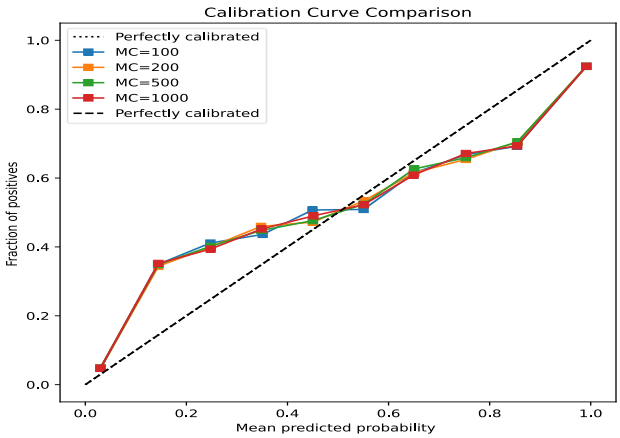


Figure 3. The calibration curve compared between different MC sample sizes.

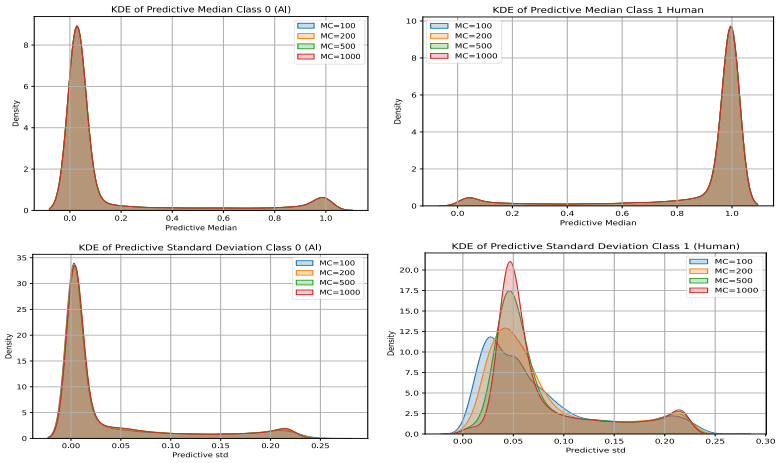


Figure 4. The boxplots show the distribution of predictive median and its S.D. in each class.4

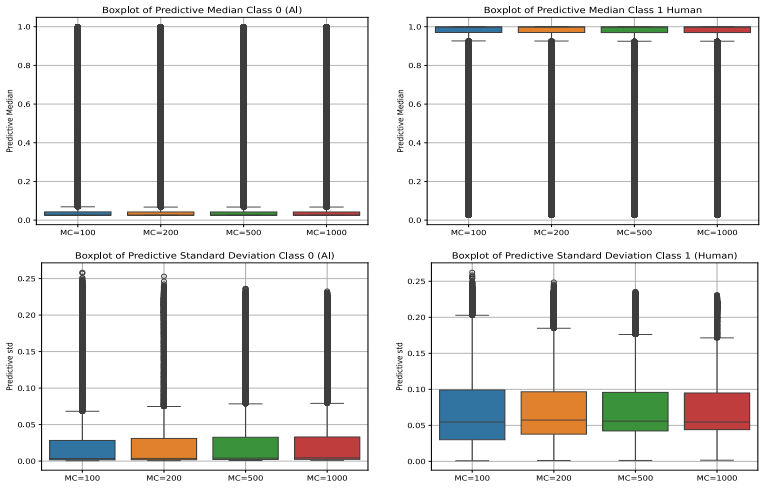


Figure 5. KDE plots illustrate the distributions of the predictive median and its S.D. in each class.

Additionally, we investigated whether the similar output distributions share the same median using the Mann-Whitney (MW) test, based on the assumption that both distributions come from the same population and share the same variance.

Hypotheses

Kolmogorov–Smirnov Test:

H_0: Two distributions are identical.

H_1: Two distributions are not identical.

Mann-Whitney test:

H_0: Two distributions share the same median.

H_1: Two distributions do not share the same median.

We considered p-values > 0.05 as failure to reject the null hypothesis (95% confidence level).

Table 4. Results for distributional and median similarity across different MC sample sizes.

MC Samples Comparison	KS similarity	Passed KS then MW similarity
Class 0 MC = 100 and MC = 200	0.9285	0.9478
Class 0 MC = 100 and MC = 500	0.9402	0.9720
Class 0 MC = 100 and MC = 1000	0.9420	0.9728
Class 0 MC = 200 and MC = 500	0.9514	0.9753
Class 0 MC = 200 and MC = 1000	0.9486	0.9709
Class 0 MC = 500 and MC = 1000	0.9649	0.9817
Class 1 MC = 100 and MC = 200	0.9285	0.9478
Class 1 MC = 100 and MC = 500	0.9402	0.9720
Class 1 MC = 100 and MC = 1000	0.9421	0.9728
Class 1 MC = 200 and MC = 500	0.9514	0.9753
Class 1 MC = 200 and MC = 1000	0.9486	0.9709
Class 1 MC = 500 and MC = 1000	0.9649	0.9817

Table 4 shows that KS similarity proportions consistently exceeded 92%, indicating that distribution shapes remained stable across MC sizes. The MW test further confirms strong agreement in median values. The highest distributional and median similarities occurred between MC = 500 and MC = 1000. We conducted a sensitivity analysis of posterior estimates with respect to the number of Monte Carlo samples; the diminishing marginal gain in stability beyond MC = 500 suggests that this value represents a practical trade-off point between computational cost and Monte Carlo error. To formally evaluate the normality of the predictive median distributions, 100 outer iterations were performed for each MC sample size. In each iteration, forward passes were made through the BNN to generate MC = 100, 200, 500, and 1000 samples; the median prediction per instance was recorded. A Shapiro-Wilk test at the 95% confidence level was then applied to the resulting distribution of median predictions across the 100 outer iterations.

Hypothesis

H_0: The distribution follows a normal distribution.

H_1: The distribution does not follow a normal distribution.

The number of predictions following a normal distribution increased with the number of MC samples, suggesting greater predictive stability. The results are summarized in Table 5.

Table 4. The results of the normality test of the sample distribution of the median across different MC sample sizes.

MC sample	Normal distribution records	Non-normal distribution records
100	35888 (59.08%)	24855 (40.92%)
200	45778 (75.36%)	14965 (24.64%)
500	52620 (86.63%)	8123 (13.37%)
1000	55274 (91.00%)	5469 (9.00%)

3.4 Prediction Hypotheses

Since the BNN produces predictive medians and predictive standard deviations, statistical hypothesis testing can be applied to its output distributions to support more nuanced decision-making beyond a fixed 0.5 threshold. A 95% confidence level was used throughout to simulate real-world error tolerance. Given that BNN outputs are not normally distributed, non-parametric statistical methods are theoretically preferred. However, parametric t-tests have been shown to perform well on large samples regardless of distributional assumptions [46]. In this experiment, both parametric and nonparametric tests were used at the 95% confidence level.

Hypotheses

Wilcoxon Test. Let $D = \text{predictive median} - 0.5$.

H_0: $D \geq 0$; assigned to class 1.

H_1: $D < 0$; assigned to class 0.

Student's t-Test. Let μ denote the mean of the output distribution.

H_0: $\mu \geq 0.5$; assigned to class 1.

H_1: $\mu < 0.5$; assigned to class 0.

Table 5. The accuracy of our model with and without statistical inferences.

Statistical Test	MC Samples	Human Recall	AI Recall	Avg Recall
No test	100	90.80%	88.37%	89.59%
	200	90.88%	88.29%	89.58%
	500	90.86%	88.28%	89.57%
	1000	90.84%	88.34%	89.59%
Wilcoxon	100	91.31%	87.79%	89.55%
	200	91.26%	87.89%	89.57%
	500	91.08%	88.05%	89.57%
	1000	91.01%	88.12%	89.56%
T-test	100	91.32%	87.79%	89.55%
	200	91.26%	87.88%	89.57%
	500	91.10%	88.05%	89.57%
	1000	91.00%	88.11%	89.56%

3.5 Discussion

In this study, we explored the effectiveness of the Bayesian Meta-Ensemble of Stacked LSTM-CNNs for AI-generated text detection. To address class imbalance in the training data, resampling was applied to create three balanced subsets and one full subset, each retaining all human-written records and using distinct partitions of the AI-generated samples for balanced training. Three base architectures: LSTM-CNN, BiLSTM-CNN, and Multi-head Attention LSTM-CNN were trained with ALBERT embeddings on each subset. The regular LSTM-CNN consistency achieved the best validation performance. This outcome aligns with findings by Zhang et al.[14] and Jang et al.[25], who similarly observed that LSTM-CNN hybrids offer strong sequence modeling with manageable complexity. The relative underperformance of BiLSTM-CNN and Multi-head Attention LSTM-CNN in this context is likely due to the parameter constraints (<6 million). The BNN meta-learner was developed by using the best-performing base models from each subset, producing probabilistic

outputs through Variational Inference. outperformed FastText, GLTR, and DetectGPT on all reported metrics and closely matched Longformer, achieving 90.84% human recall, 89.59% average recall, and 0.96 AUROC with only 18.77M parameters, approximately 7.5 times fewer than Longformer's 142M. This efficiency advantage is particularly noteworthy given that transformer-based detectors such as RoBERTa and other GPT detector variants, while achieving competitive detection rates, typically demand substantially more parameters and computational resources. The consistently higher human recall of our model relative to FastText (86.34%) and DetectGPT (86.92%) is likely due to the ensemble's ability to combine complementary feature representations across multiple training partitions, thereby reducing sensitivity to domain-specific distributional shifts in the MAGE dataset. The slightly lower average recall compared to Longformer (89.59% vs. 90.53%) reflects the inherent trade-off between model capacity and efficiency. However, the gap is marginal and offset by the substantial reduction in parameter count and the addition of uncertainty estimation.

The effect of Monte Carlo (MC) sampling size was systematically examined using Kolmogorov-Smirnov (KS) and Mann-Whitney (MW) tests to assess distributional and median consistency across MC = 100, 200, 500, and 1000. KS similarity proportions consistently exceeded 92% across all pairwise comparisons, and MW agreement in medians was similarly high, confirming that the BNN's predictive distributions are stable. Importantly, the highest similarity (KS: 0.9649, MW: 0.9817) occurred between MC = 500 and MC = 1000, indicating that marginal gains in stability diminish beyond 500 samples. The Shapiro-Wilk normality test further revealed that the proportion of predictions following a normal distribution increased substantially from 59.08% at MC = 100 to 91.00% at MC = 1000, supporting the interpretation that larger MC sample sizes allow the Central Limit Theorem to dominate, resulting in more regular, predictable output distributions. This behavior is consistent with Monte Carlo convergence theory and provides a principled basis for selecting MC = 1000 as a practical default in deployment scenarios requiring high stability.

The integration of parametric (Student's t-test) and non-parametric (Wilcoxon test) hypothesis testing into the prediction pipeline provided a modest but consistent improvement in human recall (up to 91.32% from the baseline 90.84%) while maintaining overall classification balance. Both tests yielded nearly identical results. The robustness of the t-test under non-normality is well-supported by large-sample theory and consistent with Lumley et al. and Fagerland, who demonstrated that t-tests remain valid for large samples regardless of the underlying distribution. The slight improvement in human recall from hypothesis testing suggests that incorporating distributional information from the BNN output, rather than relying solely on a fixed 0.5 threshold, improves recall. This is especially valuable in safety-critical applications such as academic integrity monitoring or misinformation detection, where errors carry higher costs. In conclusion, the Bayesian Meta-Ensemble of Stacked LSTM-CNNs for AI-generated text detection delivers competitive performance while being highly efficient. Unlike baseline models that produce only deterministic outputs, our model introduces uncertainty estimation, enabling statistical analysis to enhance reliability and interpretability in real-world applications further further. Future work should explore whether incorporating larger base models, alternative embeddings beyond ALBERT, or MCMC-based inference could further narrow the performance gap with larger transformer-based detectors, and whether the framework generalizes to multilingual or domain-shifted AI-generated text.

4. Conclusions

This study presented the Bayesian Meta-Ensemble of Stacked LSTM-CNNs, a lightweight, uncertainty-aware framework for AI-generated text detection. Using the MAGE benchmark dataset, we demonstrated that a compact LSTM-CNN trained on resampled, class-balanced subsets with ALBERT embeddings consistently outperforms more complex architectures (BiLSTM-CNN, Multi-head Attention LSTM-CNN) under a strict parameter budget (<6M), highlighting the value of architectural simplicity when model capacity is constrained. The BNN meta-learner, trained via Variational Inference on the outputs of the best-performing base models, enables probabilistic classification that goes beyond fixed-threshold decision-making. Benchmarked against FastText, GLTR, DetectGPT, and Longformer, the proposed model achieved 90.84% human recall, 89.59% average recall, and 0.96 AUROC with only 18.77M parameters, approximately

7.5 times fewer than Longformer. While RoBERTa-based and GPT-detector models represent the current state of the art in accuracy, they demand substantially more computational resources. Our model offers a competitive alternative for resource-constrained deployments. Stability analysis using Kolmogorov-Smirnov and Mann-Whitney tests confirmed that predictive distributions converge across MC sample sizes, with optimal stability observed between MC = 500 and MC = 1000. Shapiro-Wilk testing further revealed that larger MC samples yield more normally distributed prediction medians, supporting the use of parametric methods at higher sampling rates. The integration of both parametric (Student's t-test) and non-parametric (Wilcoxon) hypothesis tests into the prediction pipeline yielded a consistent improvement in human recall while preserving overall classification balance, demonstrating that leveraging the full output distribution of the BNN is more informative than relying on a fixed decision threshold alone. Overall, the proposed model advances AI-generated text detection by combining ensemble stacking, Bayesian uncertainty quantification, and statistical inference into a unified, computationally efficient framework. This combination offers capabilities unavailable in deterministic, single-model detectors, making the approach particularly suitable for high-stakes applications such as misinformation detection and academic integrity verification. However, this study has several limitations that should be acknowledged. First, the base models are intentionally lightweight (<6M parameters) due to computational resource constraints; larger base models or deeper stacked architectures may yield higher classification performance at the cost of increased inference time. Only ALBERT embeddings were evaluated; exploring alternative representations, such as RoBERTa, DeBERTa, or domain-adapted embeddings, could further improve feature extraction quality. The BNN meta-learner was trained using Variational Inference, which is a known approximation to the true posterior. Employing MCMC with adversarial sampling, as proposed by Zhang et al. [47], may improve the BNN meta-learner's accuracy and calibration. There are also several criteria to consider, such as model sensitivity analysis. Lastly, all experiments were conducted on the MAGE dataset in English. Generalizability to multilingual text, domain-shifted content, or adversarial paraphrased AI-generated text remains unverified.

5. Acknowledgments

Author Contributions: N. K. conducted the experiments, implemented the model, and contributed to writing and primary revision of the manuscript. P. S. provided guidance and contributed to reviewing and editing the manuscript.

Funding: This work was supported by Chiang Mai University under TA/RA Scholarship Program.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Kumar, P. Large language models (LLMs): Survey, technical frameworks, and future challenges. *Artif. Intell. Rev.* **2024**, *57*. https://doi.org/10.1007/s10462-024-10888-y

[2] Li, Z.; Zhang, W.; Zhang, H.; Fang, X. Global digital compact: A mechanism for the governance of online discriminatory and misleading content generation. *Int. J. Hum.-Comput. Interact.* **2025**, *41*(2), 1381–1396. https://doi.org/10.1080/10447318.2024.2314350

[3] Stokel-Walker, C. AI bot ChatGPT writes smart essays—Should professors worry? *Nature* **2022**. https://doi.org/10.1038/d41586-022-04397-7

[4] Ghosal, S. S.; Chakraborty, S.; Geiping, J.; Huang, F.; Manocha, D.; Bedi, A. S. Towards possibilities & impossibilities of AI-generated text detection: A survey. *arXiv* **2023**, arXiv:2310.15264.

[5] Solaiman, I.; et al. Release strategies and the social impacts of language models. *arXiv* **2019**, arXiv:1908.09203

[6] Zellers, R.; et al. Defending against neural fake news. In *Advances in Neural Information Processing Systems*; **2019**; Vol. 32.

[7] Jawahar, G.; Abdul-Mageed, M.; Lakshmanan, L. V. Automatic detection of machine generated text: A critical survey. *arXiv* **2020**, arXiv:2011.01314.

[8] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*; **2019**; pp 4171–4186.
[9] Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
[10] Liu, Y.; et al. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
[11] Pawlowski, N.; Brock, A.; Lee, M. C.; Rajchl, M.; Glocker, B. Implicit weight uncertainty in neural networks. *arXiv* **2017**, arXiv:1711.01297.
[12] Chen, W.; Li, B.; Zhang, R.; Li, Y. Bayesian computation in deep learning. *arXiv* **2025**, arXiv:2502.18300.
[13] Li, Y.; et al. MAGE: Machine-generated text detection in the wild. *arXiv* **2024**, arXiv:2305.13242. https://doi.org/10.48550/arXiv.2305.13242
[14] Zhang, J.; Li, Y.; Tian, J.; Li, T. LSTM-CNN hybrid model for text classification. In *Proc. IEEE IAEAC*; **2018**; pp 1675–1680. https://doi.org/10.1109/IAEAC.2018.8577620
[15] Xiao, L.; Wang, G.; Zuo, Y. Research on patent text classification based on Word2Vec and LSTM. In *Proc. ISCID*; **2018**; pp 71–74. https://doi.org/10.1109/ISCID.2018.00023
[16] Xie, J.; Chen, B.; Gu, X.; Liang, F.; Xu, X. Self-attention-based BiLSTM model for short text fine-grained sentiment classification. *IEEE Access* **2019**, *7*, 180558–180570. https://doi.org/10.1109/ACCESS.2019.2957510
[17] Wadud, M. A. H.; Kabir, M. M.; Mridha, M. F.; Ali, M. A.; Hamid, M. A.; Monowar, M. M. Managing offensive text in social media: A text classification approach using LSTM-BOOST. *Int. J. Inf. Manage. Data Insights* **2022**, *2*(2), 100095. https://doi.org/10.1016/j.jjimei.2022.100095
[18] Anggrainingsih, R.; Hassan, G. M.; Datta, A. Evaluating BERT-based language models for detecting misinformation. *Neural Comput. Appl.* **2025**. https://doi.org/10.1007/s00521-025-11101-z
[19] O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
[20] Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882. https://doi.org/10.48550/arXiv.1408.5882
[21] Alshubaily, I. TextCNN with attention for text classification. *arXiv* **2021**, arXiv:2108.01921. https://doi.org/10.48550/arXiv.2108.01921
[22] Zhao, W.; Zhu, L.; Wang, M.; Zhang, X.; Zhang, J. WTL-CNN: A news text classification method based on weighted word embedding. *Connect. Sci.* **2022**, *34*(1), 2291–2312. https://doi.org/10.1080/09540091.2022.2117274
[23] Ran, Y.; Han, H. Text classification algorithm based on sparse distributed representation. In *Proc. IEEE AEECA*; **2020**; pp 876–880. https://doi.org/10.1109/AEECA49918.2020.9213479
[24] Li, C.; Zhan, G.; Li, Z. News text classification based on improved Bi-LSTM-CNN. In *Proc. IEEE ITME*; **2018**; pp 890–893. https://doi.org/10.1109/ITME.2018.00199
[25] Jang, B.; Kim, M.; Harerimana, G.; Kang, S.; Kim, J. W. Bi-LSTM model to increase accuracy in text classification. *Appl. Sci.* **2020**, *10* (17), 5841. https://doi.org/10.3390/app10175841
[26] Abdullah, A. A.; Hassan, M. M.; Mustafa, Y. T. Bayesian deep learning in healthcare: Applications and challenges. *IEEE Access* **2022**, *10*, 36538–36562. https://doi.org/10.1109/ACCESS.2022.3163384
[27] Graves, A. Practical variational inference for neural networks.
[28] Abburi, H.; et al. A simple yet efficient ensemble approach for AI-generated text detection. In *Proc. GEM Workshop*; Association for Computational Linguistics: Singapore, 2023; pp 413–421.
[29] Aggarwal, K.; Singh, S.; Parul; Pal, V.; Yadav, S. S. Enhancing accuracy in AI-generated text detection using ensemble modelling. In *Proc. IEEE TENSYMP*; 2024; pp 1–8. https://doi.org/10.1109/TENSYMP61132.2024.10752173
[30] Massey, F. J., Jr. The Kolmogorov–Smirnov test for goodness of fit. *J. Am. Stat. Assoc.* **1951**, *46* (253), 68–78. https://doi.org/10.1080/01621459.1951.10500769
[31] Two Sample Kolmogorov–Smirnov. Real Statistics Using Excel. https://real-statistics.com/non-parametric-tests/goodness-of-fit-tests/two-sample-kolmogorov-smirnov-test/ (accessed Jul 23, 2025).
[32] Divine, G. W.; Norton, H. J.; Barón, A. E.; Juarez-Colunga, E. The Wilcoxon–Mann–Whitney procedure fails as a test of medians. *Am. Stat.* **2018**, *72*(3), 278–286.

[33] Feltovich, N. Nonparametric tests of differences in medians. *Exp. Econ.* **2003**, *6*(3), 273–297.
[34] Yue, S.; Wang, C. Y. Power of the Mann–Whitney test. *Stochastic Environ. Res. Risk Assess.* **2002**, *16*(4), 307–323.
[35] Mishra, P.; Singh, U.; Pandey, C. M.; Mishra, P.; Pandey, G. Application of Student's t-test. *Ann. Card. Anaesth.* **2019**, *22*(4), 407–411.
[36] Lumley, T.; Diehr, P.; Emerson, S.; Chen, L. Importance of normality assumption. *Annu. Rev. Public Health* **2002**, *23*, 151–169.
[37] Ratcliffe, J. F. Effect on the t distribution of non-normality. *Appl. Stat.* **1968**, *17*(1), 42.
[38] Sawilowsky, S. S.; Blair, R. C. Robustness of the t test. *Psychol. Bull.* **1992**, *111*(2), 352–360.
[39] Althouse, L. A.; Ware, W. B.; Ferron, J. M. Detecting departures from normality. **1998**.
[40] Razali, N. M.; Yap, B. Power comparisons of normality tests. *J. Stat. Model. Anal.* **2011**, *2*.
[41] UMA Technology. The dangers of AI writing and how to spot AI-generated text. https://umatechnology.org/the-dangers-of-ai-writing-and-how-to-spot-ai-generated-text/ (accessed Mar 14, 2025).
[42] AI's Jurassic Park moment. *Commun. ACM*. https://cacm.acm.org/blogcacm/ais-jurassic-park-moment/ (accessed Mar 14, 2025).
[43] Vincent, J. AI-generated answers temporarily banned on Stack Overflow. *The Verge*. https://www.theverge.com/ (accessed Mar 14, 2025).
[44] Dergaa, I.; Chamari, K.; Zmijewski, P.; Saad, H. B. AI-generated text in academic writing. *Biol. Sport* **2023**, *40*(2), 615–622.
[45] Bohannon, J. Who's afraid of peer review? *Science* **2013**, *342*(6154), 60–65.
[46] Fagerland, M. W. t-tests and non-parametric tests. *BMC Med. Res. Methodol.* **2012**, *12*, 78.
[47] Zhang, J.; et al. Improving Bayesian neural networks by adversarial sampling. *Proc. AAAI* **2022**, *36*(9), 10110–10117.