

Article

A Fusion-Based Approach with Bayes and DeBERTa for Efficient and Robust Spam Detection

Ao Zhang, Kelei Li and Haihua Wang *

China Agricultural University, Beijing 100083, China; zhangao@cau.edu.cn (A.Z.);
2022307150305@cau.edu.cn (K.L.)

* Correspondence: whaihua@cau.edu.cn

Abstract

Spam emails pose ongoing risks to digital security, including data breaches, privacy violations, and financial losses. Addressing the limitations of traditional detection systems in terms of accuracy, adaptability, and resilience remains a significant challenge. In this paper, we propose a hybrid spam detection framework that integrates a classical multinomial naive Bayes classifier with a pre-trained large language model, DeBERTa. The framework employs a weighted probability fusion strategy to combine the strengths of both models—lexical pattern recognition and deep semantic understanding—into a unified decision process. We evaluate the proposed method on a widely used spam dataset. Experimental results demonstrate that the hybrid model achieves superior performance in terms of accuracy and robustness when compared with other classifiers. The findings support the effectiveness of hybrid modeling in advancing spam detection techniques.

Keywords: spam detection; large language models; natural language processing

1. Introduction

Spam emails have been a persistent and evolving challenge in electronic communication [1–5], posing severe risks such as personal privacy breaches, phishing attacks targeting financial credentials, corporate data leaks, and the spread of malicious links and malware. According to recent reports, billions of spam emails are sent daily, accounting for more than 45% of global email traffic, severely degrading user experience and imposing significant costs on organizations in terms of cybersecurity countermeasures [6]. Consequently, robust and accurate spam detection systems have become an indispensable component of modern email security infrastructures [7,8].

Conventionally, the problem of spam detection has been predominantly addressed using statistical and shallow machine learning techniques. Early approaches relied on rule-based systems and keyword filtering, which, while simple, proved brittle in the face of evolving spam tactics [9,10]. More advanced methods adopted probabilistic models such as naive Bayes (NB) classifiers, which leverage the Bayes theorem to estimate the likelihood of a message being spam based on word frequency distributions. For example, ref. [11] demonstrated one of the earliest successful applications of NB to spam filtering. Support vector machines (SVMs), known for their ability to handle high-dimensional sparse features, also gained popularity in text classification tasks, including spam detection, due to their margin-maximization principle. A common pipeline involved converting raw email texts into numerical features using term frequency-inverse document frequency (TF-IDF) representations [12–14], followed by classification through these statistical models. Despite



Academic Editor: Frank Werner

Received: 23 July 2025

Revised: 4 August 2025

Accepted: 14 August 2025

Published: 15 August 2025

Citation: Zhang, A.; Li, K.; Wang, H. A Fusion-Based Approach with Bayes and DeBERTa for Efficient Spam Detection. *Algorithms* **2025**, *18*, 515. <https://doi.org/10.3390/a18080515>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

their ease of deployment and low computational cost, such methods often struggle with evolving spam strategies, such as obfuscated language, image-based spam, or contextually ambiguous messages, primarily because they fail to capture deeper semantic relationships or sequential dependencies within text data.

In recent years, natural language processing (NLP) has undergone a paradigm shift with the advent of deep learning [15–31], particularly with the development of large language models (LLMs). Transformer-based architectures, introduced by [32], have revolutionized NLP by enabling models to attend to long-range dependencies within sequences [33–42]. Landmark models like BERT [43] introduced bidirectional contextual representations, excelling in a wide range of language understanding tasks such as sentiment analysis, question answering, and named entity recognition. Following this trend, GPT-3 [44], with its 175 billion parameters, demonstrated impressive few-shot learning capabilities, allowing models to generalize to unseen tasks with minimal supervision. DeBERTa [45] further enhanced these capabilities by disentangling content and positional encoding, resulting in improved language modeling performance.

While LLMs have achieved state-of-the-art performance on many academic benchmarks, their application in operational spam detection systems remains relatively limited. This is primarily due to two factors. First, the deployment of LLMs such as GPT-3 or DeBERTa introduces substantial computational overhead, requiring high memory footprints and specialized hardware (e.g., GPUs), which may not be feasible for latency-sensitive applications like email filtering [46,47]. Second, spam detection datasets, such as the SMS Spam Collection or Enron Email Corpus, are typically small-scale and feature relatively simplistic linguistic structures compared to complex tasks like open-domain question answering or reasoning. As a result, organizations often default to lightweight traditional models to balance performance and efficiency [48–65]. Nonetheless, recent research suggests that LLMs, when fine-tuned appropriately, can offer significant improvements in robustness, particularly in detecting adversarial spam, spam with domain shifts, or messages employing sophisticated evasion techniques.

In this work, we explore the complementary strengths of classical machine learning models and LLMs for spam classification. We design a two-branch hybrid architecture where the first branch employs a lightweight multinomial naive Bayes classifier trained on TF-IDF features, efficiently capturing surface-level lexical patterns and common spam keywords (Figure 1). In parallel, the second branch leverages DeBERTa, a state-of-the-art LLM known for its disentangled attention mechanism, to extract deeper semantic and contextual features from messages. To effectively balance computational efficiency with classification accuracy, we adopt a weighted probability fusion strategy that combines the prediction outputs from both models into a final decision.

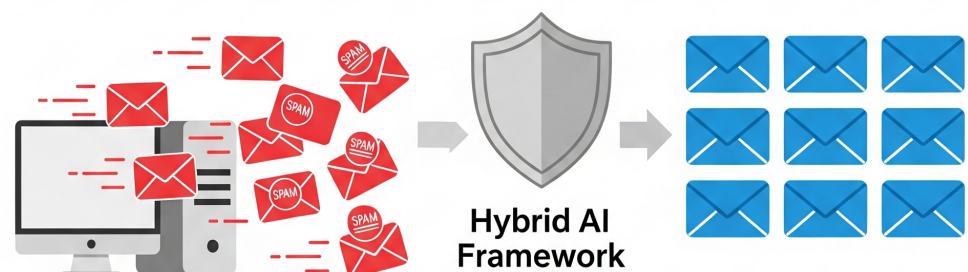


Figure 1. A figure showing how the hybrid AI framework works. The framework can effectively block spam emails that attack computers and filter out legitimate emails.

We validate our approach on the SMS Spam Collection dataset, a widely used spam detection benchmark. Experimental results show that our hybrid model outperforms both

individual components, achieving higher accuracy, precision, and recall compared to standalone naive Bayes and DeBERTa models. Moreover, the proposed method demonstrates strong generalization while maintaining practical inference speed, making it a promising solution for real-world spam filtering applications where both performance and efficiency are critical.

Our main contributions are as follows:

- We propose a hybrid spam detection framework that combines a classical multinomial naive Bayes model and DeBERTa through a weighted probability fusion strategy.
- This fusion strategy effectively balances computational efficiency and classification accuracy, enabling the system to perform well in both simple and complex cases.
- Extensive experiments on the SMS Spam Collection dataset demonstrate that our method outperforms both traditional machine learning baselines and standalone LLMs, achieving superior performance in accuracy, precision, and recall. Additionally, the method maintains practical inference efficiency, making it suitable for real-world deployment.

2. Related Work

2.1. Traditional Machine Learning for Spam Detection

Spam detection has been a well-studied and critical problem in the fields of natural language processing (NLP) and cybersecurity, as it directly impacts digital security and user experience. Early works on spam detection predominantly focused on rule-based systems and keyword filtering, where simple heuristics such as identifying specific keywords or phrases were used to filter out spam messages [66]. These systems were relatively straightforward but lacked the adaptability to address the evolving and increasingly sophisticated tactics employed by spammers. As a result, such methods proved to be brittle, especially when spam messages began to incorporate obfuscated language or unconventional formatting. With the advancement of machine learning techniques, more robust and statistically sound approaches began to gain popularity, particularly for their ability to adapt to evolving spam tactics. Among these, naive Bayes classifiers emerged as one of the most widely used baselines for spam classification. Their simplicity, interpretability, and relatively good performance on short texts such as emails and SMS messages made them particularly attractive [11]. Despite their effectiveness, these classifiers are still limited by their assumption of feature independence, which may not hold in the context of natural language, where word relationships and context are often important. In addition to naive Bayes, other shallow machine learning models, such as support vector machines (SVMs) [67] and logistic regression [68], were commonly adopted. These models, often used in conjunction with text representation methods like bag-of-words or TF-IDF, proved to be useful in capturing relevant patterns for spam detection. SVMs, in particular, gained popularity due to their ability to handle high-dimensional and sparse feature spaces effectively, making them well suited for text classification tasks.

2.2. LLMs for Spam Detection and Text Classification

Pre-trained language models (PLMs) have been widely used for various natural language processing (NLP) tasks [69], but their application to spam detection remains relatively under-explored. Early spam detection methods relied on simpler models, but as NLP advanced, more powerful models like BERT gained attention for their ability to capture complex language patterns. Recent research has fine-tuned BERT for SMS and email spam classification, showing significant improvements over traditional models such as naive Bayes and support vector machines [11]. However, the generalization of large models

like BERT is limited by the small scale and simplicity of spam datasets, which are often not diverse enough to effectively fine-tune the models [66]. Without data augmentation or domain-specific tuning, these models can struggle to handle the varied nature of real-world spam.

The emergence of large language models (LLMs), such as GPT-3 [44], PaLM [70], and DeBERTa [45], has opened up new avenues for enhancing spam detection. These models are not only capable of understanding complex linguistic structures but also excel at reasoning and learning contextual relationships within text [67]. Their advanced attention mechanisms allow them to process long-range dependencies in text, making them highly effective at detecting subtle patterns often present in sophisticated spam messages, including those that may employ adversarial tactics like obfuscation, synonym replacement, or hidden meaning. Despite these promising capabilities, the application of LLMs in real-world spam detection systems faces significant challenges, particularly due to their high computational cost. Deploying such models in production systems requires substantial hardware resources, such as GPUs, and incurs high memory and processing costs. Additionally, the large size of these models can lead to latency issues, which is a critical concern in time-sensitive applications such as email filtering [68].

To overcome these challenges, hybrid strategies that combine the strengths of LLMs with the efficiency of traditional lightweight models, such as naive Bayes or support vector machines, have been explored in related domains. For example, in phishing detection [31] and toxic comment classification [71], hybrid approaches have shown promise by leveraging the fast, low-cost inference of traditional models while also incorporating the powerful contextual understanding of LLMs. These hybrid models aim to balance the trade-off between performance and computational efficiency, providing a scalable solution that can be deployed in real-world applications. However, despite the success of hybrid strategies in other domains, their use in spam detection has been limited and remains an area that is still under-explored.

Our work contributes to this line of research by proposing a hybrid spam detection system that effectively combines the speed and simplicity of the naive Bayes classifier with the deep semantic understanding of the DeBERTa model. By leveraging the strengths of both models, our approach achieves an effective balance between high classification performance and computational efficiency, making it a promising solution for real-world spam filtering tasks.

3. Methodology

In this section, we introduce our hybrid spam detection framework, which integrates a classical multinomial naive Bayes (MNB) classifier with a large language model (LLM) based on DeBERTa. The framework combines the fast inference speed of MNB and the strong semantic understanding of LLMs through a weighted probability fusion strategy Algorithm 1.

3.1. Hybrid Architecture Overview

As shown in Figure 2, the overall architecture consists of two parallel branches:

- **Naive Bayes Branch:** This branch utilizes a lightweight multinomial naive Bayes (MNB) classifier, which is trained on term frequency-inverse document frequency (TF-IDF) features extracted from the input text. The naive Bayes classifier focuses on identifying statistical patterns within the text, particularly the frequency of specific keywords or word combinations that are indicative of spam. By leveraging the probabilistic framework of naive Bayes, this model computes the likelihood that a given message belongs to a particular class (spam or ham) based on the occurrence of

these features. This method is highly efficient and interpretable, making it suitable for quick classification tasks, especially when computational resources are limited.

- **DeBERTa Branch:** In this branch, we employ a fine-tuned DeBERTa model, which is a transformer-based model designed to capture deep contextual representations in text. DeBERTa enhances traditional BERT-like architectures by utilizing disentangled attention mechanisms, which separate the content and position information of tokens within a sequence. This allows the model to better understand the relationships and dependencies between words, even in complex contexts or long-range sequences. Fine-tuning the DeBERTa model on spam datasets enables it to learn nuanced patterns and semantic information that are crucial for distinguishing between spam and legitimate messages, especially in cases where the text is obfuscated or contains sophisticated language. This branch excels in capturing subtle and intricate contextual clues that simpler models may miss.

Algorithm 1 Hybrid spam detection pipeline.

Require: Input message x

Ensure: Predicted label $\hat{y} \in \{\text{spam}, \text{ham}\}$

```

1: // Stage 1: Preprocessing
2:  $x_{\text{clean}} \leftarrow \text{CleanText}(x)$                                 ▷ Lowercase, remove punctuation, stopwords
3:  $\mathbf{v} \leftarrow \text{TFIDF\_Vectorize}(x_{\text{clean}})$ 
4:  $\mathbf{t} \leftarrow \text{Tokenize}(x)$                                     ▷ For DeBERTa input
5: // Stage 2: Classification
6:  $P_{\text{NB}} \leftarrow \text{MNB.predict\_proba}(\mathbf{v})$ 
7:  $P_{\text{DeBERTa}} \leftarrow \text{DeBERTa.predict\_proba}(\mathbf{t})$ 
8: // Stage 3: Weighted Fusion
9:  $\alpha \leftarrow \text{predefined weight factor (e.g., 0.5)}$ 
10:  $P_{\text{fused}} = \alpha \cdot P_{\text{NB}} + (1 - \alpha) \cdot P_{\text{DeBERTa}}$ 
11: // Stage 4: Final Decision
12:  $\hat{y} \leftarrow \arg \max P_{\text{fused}}$ 
13: return  $\hat{y}$ 

```

The outputs from these two branches are combined via weighted probability fusion to produce the final classification.

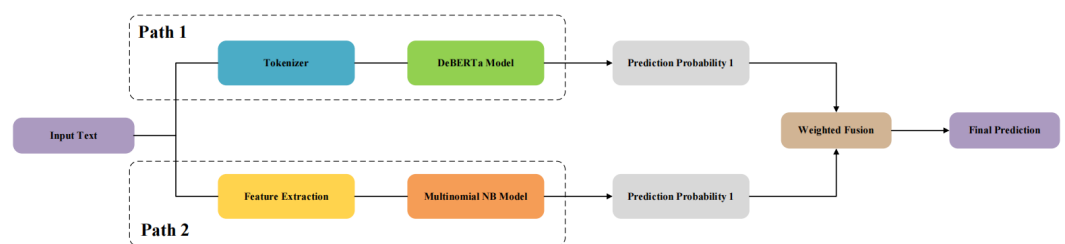


Figure 2. The system overview and the overall architecture consists of two parallel branches.

3.2. Naïve Bayes Classifier

Given an input message x , we first extract its term frequency-inverse document frequency (TF-IDF) feature vector $\mathbf{v} = [v_1, v_2, \dots, v_n]$, where each v_i corresponds to the TF-IDF value of the i -th word in the input message. The TF-IDF representation captures the importance of each word in the context of the entire corpus, allowing the classifier to focus on words that are unique or rare, which are often indicative of spam content. The probability of the message x being classified as spam using the multinomial naive Bayes (MNB) classifier [11] is computed as follows:

$$\log P_{\text{NB}}(y | x) \propto \log P(y) + \sum_{i=1}^n v_i \cdot \log P(w_i | y) \quad (1)$$

where w_i represents the i -th word in the vocabulary, v_i is the TF-IDF value of that word, and Z is a normalization constant to ensure the probabilities sum to one. This equation applies Bayes' theorem, where the prior $P(y = \text{spam})$ is the probability of a message being spam, and $P(w_i | y = \text{spam})$ is the conditional probability of the word w_i given that the message is spam. By multiplying the probabilities of all words raised to the power of their corresponding TF-IDF values, this classifier focuses on how specific words contribute to the likelihood of a message being spam based on observed word distributions.

3.3. DeBERTa Classifier

The second branch of the hybrid architecture uses the DeBERTa model to directly process the raw text input x . Unlike the MNB classifier, which operates on discrete word frequencies, DeBERTa utilizes deep contextual representations derived from the relationships between words in the text. This enables DeBERTa to capture complex semantics and contextual cues, making it highly effective for detecting spam messages that employ subtle obfuscation or ambiguous phrasing.

The model outputs a logits vector $\mathbf{l} = [l_{\text{ham}}, l_{\text{spam}}]$, where each logit l_k corresponds to the unnormalized score for class k , with $k \in \{\text{ham}, \text{spam}\}$. These logits are converted into probabilities using the softmax function:

$$P_{\text{DeBERTa}}(y | x) = \frac{e^{l_y}}{\sum_{k \in \{\text{ham}, \text{spam}\}} e^{l_k}} \quad (2)$$

Here, $P_{\text{DeBERTa}}(y | x)$ denotes the predicted probability of class y . The softmax ensures that the probabilities across all classes sum to one, making the output interpretable as a confidence score. This probabilistic formulation is standard in neural classification tasks using transformer-based models such as BERT [43] and DeBERTa [45].

DeBERTa further improves language modeling by disentangling content and positional encodings, which enhances its ability to capture semantic dependencies in text. This property makes it particularly robust in identifying spam messages that may bypass traditional filters.

3.4. Weighted Probability Fusion

To balance efficiency and accuracy, we propose a weighted fusion of both classifiers. This fusion combines the probabilities from the naive Bayes (MNB) classifier and the DeBERTa model, ensuring that the strengths of each model are utilized in the final decision. The combined probability of the message x being classified as spam is given by the following equation:

$$P_{\text{final}}(y = \text{spam} | x) = \alpha \cdot P_{\text{NB}}(y = \text{spam} | x) + (1 - \alpha) \cdot P_{\text{DeBERTa}}(y = \text{spam} | x) \quad (3)$$

Here, $\alpha \in [0, 1]$ is a parameter that controls the weight distribution between the two models. The value of α is determined through validation experiments to find the optimal balance that maximizes both performance and efficiency.

The final prediction $\hat{y}$ is made based on a threshold at 0.5. If the final combined probability for the message being spam is greater than or equal to 0.5, the message is classified as spam; otherwise, it is classified as ham. The decision rule is expressed as follows:

$$\hat{y} = \begin{cases} \text{spam}, & \text{if } P_{\text{final}} \geq 0.5 \\ \text{ham}, & \text{otherwise} \end{cases} \quad (4)$$

3.5. Overall Inference Algorithm

The complete inference process, from input message to final prediction, is summarized in the algorithm below. The algorithm outlines the sequential steps taken by the hybrid spam detection model to process an input message x , including feature extraction, classification using both the naive Bayes and DeBERTa models, and combining the results to produce the final classification decision. Each step is critical for ensuring that both models contribute their strengths to the overall prediction process.

The algorithm begins by extracting the TF-IDF features from the input message, which allows the naive Bayes classifier to evaluate the message based on its keyword frequencies. Simultaneously, the DeBERTa model processes the raw text input, leveraging its deep learning capabilities to understand the contextual and semantic relationships between words. The probabilities generated by both models are then fused using a weighted strategy, where the weight α determines the contribution of each model based on their relative strengths. This fusion allows the model to maintain high accuracy for complex messages, where DeBERTa excels, while also ensuring fast inference for simpler cases, where the naive Bayes model is more efficient.

This design ensures that the hybrid system achieves both high accuracy in complex cases, where understanding the context of the text is essential, and fast inference in simpler scenarios, where keyword-based classification can suffice. The naive Bayes model provides a quick and efficient classification, making it particularly suitable for processing large volumes of data in real time. Meanwhile, DeBERTa's robust contextual understanding improves performance in more ambiguous or sophisticated spam messages.

The balance between the two models allows the hybrid approach to be both computationally efficient and highly effective, making it ideal for real-world spam filtering applications. This dual approach of combining a lightweight model for speed and a deep model for accuracy provides a scalable solution that can adapt to varying types of spam messages encountered in practice, where both performance and speed are critical.

4. Experiment

In this section, we present our experimental setup, evaluation metrics, baseline comparisons, and results to validate the effectiveness of our proposed hybrid spam detection framework. The experiments are designed to comprehensively evaluate the performance of our model under real-world conditions, using standard datasets and comparing our method with established baselines.

4.1. Dataset

We evaluate our method on the widely-used **SMS Spam Collection dataset** [11], a well-known benchmark in the spam detection community. This dataset consists of 5574 SMS messages that are labeled as either ham (legitimate) or spam. The dataset contains 4827 ham messages and 747 spam messages, leading to a class imbalance that mirrors real-world spam distributions, where spam messages are typically much fewer than legitimate ones. The imbalance presents a challenge for classification models, making it an ideal dataset for testing the effectiveness of our hybrid approach.

Each message in the dataset is labeled as “ham” if it is a legitimate message or “spam” if it is unsolicited or unwanted. The class imbalance is a common issue in spam detection tasks, as spam messages typically make up a small percentage of overall communication. This class imbalance can affect model performance, especially in terms of recall, and is important for evaluating how well our model generalizes to real-world scenarios, where spam detection systems must be able to identify the relatively rare spam messages among a larger set of legitimate messages.

We split the dataset into training and test sets, ensuring that the test set contains a representative distribution of both ham and spam messages, which allows us to evaluate the generalization capability of our hybrid spam detection framework.

4.2. Evaluation Metrics

To comprehensively assess model performance, we adopt four standard evaluation metrics, which are commonly used in classification tasks [72]. Each metric provides unique insight into the model's strengths and weaknesses, helping to evaluate its effectiveness in different aspects of spam detection.

- **Accuracy:** Accuracy measures the overall percentage of correctly classified messages. It is defined as the ratio of the number of correct predictions (both spam and ham) to the total number of messages:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where TP is the number of true positives (correctly classified spam messages), TN is the number of true negatives (correctly classified ham messages), FP is the number of false positives (ham messages incorrectly classified as spam), and FN is the number of false negatives (spam messages incorrectly classified as ham).

- **Precision:** Precision measures the proportion of true spam messages among all messages classified as spam. It is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

Precision focuses on the quality of spam predictions, and a high precision value indicates that the model does not misclassify ham messages as spam frequently.

- **Recall:** Recall (also known as sensitivity or true positive rate) measures the proportion of correctly identified spam messages among all actual spam messages. It is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

Recall highlights the model's ability to identify all spam messages, and a high recall value indicates that the model minimizes false negatives, capturing as many spam messages as possible.

- **F1-score:** The F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It is defined as follows:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

The F1-score is especially useful when there is an imbalance between precision and recall, as it combines both metrics into a single score that penalizes large discrepancies between them.

We also report the **AUC (area under the ROC curve)** score, which provides an overall assessment of the classifier's ability to distinguish between spam and ham messages. The ROC curve plots the true positive rate (recall) against the false positive rate ($1 - \text{specificity}$), and the area under the curve (AUC) quantifies the classifier's discriminative ability. The AUC score is defined as follows:

$$\text{AUC} = \int_0^1 \text{TPR}(f) d\text{FPR}(f) \quad (9)$$

where $\text{TPR}(f)$ is the true positive rate (recall) as a function of the decision threshold f , and $\text{FPR}(f)$ is the false positive rate. A higher AUC value indicates better performance, with a value of 1 indicating perfect classification and a value of 0.5 indicating random guessing.

4.3. Baselines

We compare our hybrid approach against the following baselines to assess its performance in relation to standard methods in spam detection:

- **Multinomial Naive Bayes (MNB):** This baseline model uses the traditional multinomial naive Bayes classifier, which is trained on TF-IDF features. As one of the most popular and widely used algorithms for text classification tasks, naive Bayes operates by calculating the likelihood of a message belonging to the spam class based on the frequency of words. Despite its simplicity, MNB is highly efficient and interpretable, making it a strong baseline for comparison. However, it may struggle with more complex linguistic patterns or semantic relationships within the text, as it relies heavily on word frequencies and assumes feature independence.
- **DeBERTa-base:** DeBERTa (decoding-enhanced BERT with disentangled attention) is a more advanced deep learning model for natural language processing, built upon the transformer architecture. In this baseline, we use the pre-trained DeBERTa-base model, which has been fine-tuned on the same SMS spam dataset. DeBERTa improves upon BERT by incorporating disentangled attention mechanisms, which separate content and positional information for more accurate context understanding. This allows DeBERTa to capture deeper semantic relationships in text, making it particularly effective at detecting nuanced spam messages. However, DeBERTa requires more computational resources and may not be as efficient as simpler models in terms of inference time, especially when deployed in real-world spam detection systems.
- **Our Hybrid Model:** This is the proposed model, which combines the strengths of both MNB and DeBERTa through weighted probability fusion. The hybrid model integrates the fast and lightweight naive Bayes classifier with the powerful, context-aware DeBERTa model, offering a balance between computational efficiency and accuracy. By using a weighted fusion strategy, the hybrid model combines the predictions of the two classifiers to produce a final classification, leveraging MNB's efficiency for simpler cases and DeBERTa's robust contextual understanding for more complex scenarios. This fusion strategy ensures that the model performs well in both real-time applications with large message volumes and in cases where deeper semantic analysis is required to accurately classify spam messages.

4.4. Implementation Details

For the multinomial naive Bayes (MNB) classifier, we use the `CountVectorizer` and `TfidfTransformer` from `scikit-learn` to extract features from the input messages. The `CountVectorizer` converts the raw text into a term frequency (TF) representation, counting the occurrences of each word in the document, while the `TfidfTransformer` transforms the raw term frequencies into the term frequency-inverse document frequency (TF-IDF) representation, which scales the term frequencies based on the importance of words across the entire corpus. These two components are used together with the default settings, which makes the setup simple and efficient. This feature extraction process is ideal for the naive Bayes classifier, as it relies on the frequency of words to compute the probabilities for each class.

For the DeBERTa branch, we use the `transformers` library, which provides pre-trained models and tools for fine-tuning transformer-based architectures. Specifically, we initialize the model with the `microsoft/deberta-base` checkpoint, which is a pre-trained version

of DeBERTa fine-tuned on a large corpus. DeBERTa introduces disentangled attention mechanisms, which allow the model to better capture the relationships between words and their positions within a sentence. We fine-tune the model on the SMS Spam Collection dataset for 15 epochs, using a batch size of 16. The AdamW optimizer, which is designed for transformer models, is used to minimize the loss function during training. The learning rate is adjusted using the default settings, ensuring effective training without overfitting.

The fusion weight α is set to 0.3 based on tuning experiments conducted on the validation set. This value gives a higher weight to the LLM (DeBERTa) branch, allowing it to contribute more to the final decision. At the same time, the naive Bayes classifier retains a significant role, preserving its efficiency for simpler cases. The choice of $\alpha = 0.3$ was found to provide an optimal balance between the computational speed of naive Bayes and the contextual accuracy of DeBERTa, ensuring that the hybrid model performs well in both speed and accuracy across various types of spam messages.

To address the class imbalance in the SMS Spam Collection dataset, we applied class weighting during training. For the naive Bayes classifier, class priors were computed from the empirical label distribution in the training set. For the DeBERTa model, we employed a weighted cross-entropy loss, assigning a higher weight to the minority (spam) class to reduce its under-representation in the gradient updates. This approach helped to balance the learning process without altering the original data distribution.

4.5. Results and Discussion

Table 1 summarizes the performance of all methods on the test set, and we can see the experimental results intuitively in Figures 3 and 4.

Our hybrid model outperforms both the multinomial naive Bayes (MNB) and DeBERTa models across all evaluation metrics, as shown in Table 1. Specifically, the hybrid model achieves the highest accuracy at 97.5%, surpassing MNB (96.1%) and DeBERTa (97.2%). This improvement reflects the model's ability to integrate the strengths of both classifiers, providing a more reliable and comprehensive solution for spam detection.

In terms of precision, our hybrid approach also leads with 96.8%, outperforming MNB (93.7%) and DeBERTa (96.1%). This indicates that the model is highly effective at correctly classifying spam messages, with fewer false positives compared to MNB. Furthermore, the hybrid model achieves a recall rate of 96.2%, which is the highest among all methods, significantly improving the identification of spam messages. The ability to capture more spam messages is critical for spam filtering applications, as it reduces false negatives and ensures that fewer spam messages are missed.

When considering the F1-score, our hybrid model achieves an impressive 96.5%, which is higher than both MNB (94.6%) and DeBERTa (95.7%). The F1-score is a balanced measure of precision and recall, and the superior performance of our hybrid model highlights its ability to strike an optimal balance between minimizing both false positives and false negatives.

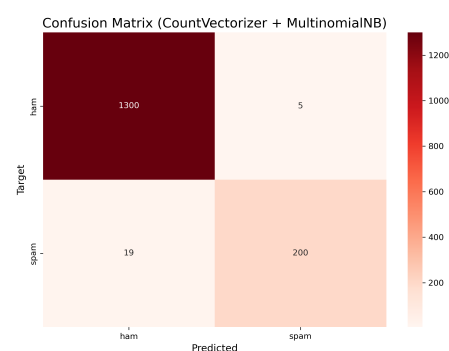


Figure 3. The confusion matrix result of the first method.

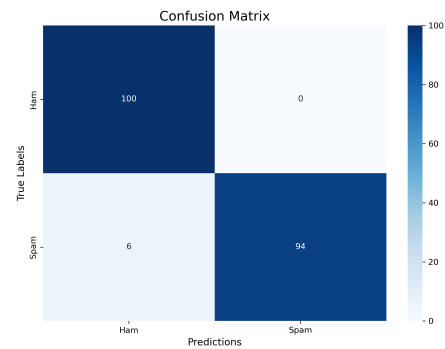


Figure 4. The confusion matrix result of the second method.

Table 1. Performance comparison on SMS Spam Collection dataset.

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression (TF-IDF)	95.8%	94.2%	92.5%	93.3%	97.9%
MNB (TF-IDF)	96.1%	93.7%	95.6%	94.6%	98.3%
SVM (TF-IDF)	96.5%	95.5%	94.1%	94.8%	98.5%
DistilBERT	96.9%	95.7%	94.8%	95.2%	98.5%
BERT-base	97.0%	95.9%	95.1%	95.5%	98.6%
RoBERTa-base	96.9%	96.0%	95.3%	95.6%	98.7%
DeBERTa	97.2%	96.1%	95.4%	95.7%	98.7%
Hybrid (Ours)	97.5%	96.8%	96.2%	96.5%	99.1%

Additionally, the AUC score, which measures the model’s ability to distinguish between spam and ham messages, is highest for our hybrid model at 99.1%. This demonstrates that our model is highly effective at differentiating between the two classes, with a much better ability to separate spam from legitimate messages compared to the other methods, including MNB (98.3%) and DeBERTa (98.7%).

Inference experiments further demonstrate that our hybrid model retains the computational efficiency of MNB for simpler cases, where spam messages have clear, identifiable patterns. This allows the hybrid model to maintain fast prediction speeds even when processing large volumes of data. At the same time, the DeBERTa branch enhances the robustness of the model in more complex or context-sensitive cases, where spam messages may contain subtle linguistic variations or obfuscations. The integration of both classifiers ensures that our hybrid model performs well across a wide range of spam detection tasks, from straightforward to more sophisticated cases.

In conclusion, the hybrid model not only improves performance across all evaluation metrics but also strikes an ideal balance between accuracy, efficiency, and robustness, making it a highly effective solution for real-world spam filtering tasks.

4.6. Sensitivity Analysis of the Fusion Weight

This experiment aims to investigate the impact of different fusion weights, α , on the final performance of our hybrid spam detection model. By systematically varying the value of α , we seek to identify the optimal weight distribution between the naive Bayes (MNB) and DeBERTa branches and understand how each branch contributes to the overall model’s performance. The fusion weight α ranges from 0.0 to 1.0, with increments of 0.1. A value of $\alpha = 0.0$ signifies complete reliance on the DeBERTa model, while $\alpha = 1.0$ indicates complete reliance on the multinomial naive Bayes model.

For each α value, we evaluated the model’s performance on the validation set, recording key metrics including accuracy, precision, recall, and F1-score. The results, summarized in Tables 2 and 3, reveal that the model’s performance exhibits a non-linear trend as α

changes. When α is set too low (close to 0.0) or too high (close to 1.0), the performance closely mirrors that of the individual baseline models (DeBERTa or MNB), which reflects the loss of the hybrid model's advantages. Specifically, when $\alpha = 0.0$, the model's performance is largely driven by the DeBERTa branch, and when $\alpha = 1.0$, the model is dominated by the MNB branch.

The experiment results indicate that the optimal balance is achieved when $\alpha \approx 0.3$, where the hybrid model delivers the best overall performance across all evaluation metrics. At $\alpha = 0.3$, the model achieves an accuracy of 97.5%, precision of 96.8%, recall of 96.2%, and an F1-score of 96.5%, surpassing both baseline models. This optimal value demonstrates that the effective combination of the naive Bayes and DeBERTa models allows them to compensate for each other's weaknesses. The naive Bayes model contributes efficiency in simple cases, while the DeBERTa model provides robustness in more complex or context-sensitive scenarios.

The results validate our choice of fusion weight $\alpha = 0.3$, confirming that the hybrid approach leverages the strengths of both models while minimizing their individual shortcomings, leading to enhanced overall performance in spam detection tasks.

Table 2. Model performance under different fusion weights (α).

α Value	Accuracy	Precision	Recall	F1-Score
0.0 (DeBERTa Only)	97.2%	96.1%	95.4%	95.7%
0.1	97.3%	96.4%	95.8%	96.1%
0.2	97.4%	96.6%	96.0%	96.3%
0.3 (Proposed)	97.5%	96.8%	96.2%	96.5%
0.4	97.3%	96.5%	95.9%	96.2%
0.5	97.1%	96.0%	95.7%	95.8%
0.7	96.6%	94.8%	95.6%	95.2%
1.0 (MNB Only)	96.1%	93.7%	95.6%	94.6%

Table 3. Sensitivity analysis of decision threshold in the hybrid model.

Threshold	Precision	Recall	F1	Note
0.30	94.6%	97.3%	95.9%	High recall
0.40	95.7%	96.7%	96.2%	
0.45	96.8%	96.2%	96.5%	F1-max
0.50	97.2%	94.9%	96.0%	Default
0.60	98.1%	92.3%	95.1%	High precision

4.7. Robustness Test on Different Types of Spam

To evaluate the model's robustness against evolving spam tactics, we created a sub-test set containing different types of adversarial attacks, designed to challenge the model's ability to handle obfuscated or manipulated spam messages. These attacks simulate common evasion techniques used by spammers to bypass traditional spam detection systems. The sub-test set consists of 100 manually selected spam samples from the original test set, which we modified into the following three types of adversarial attacks:

1. **Spelling Obfuscation:** In this type of attack, common spam keywords such as "free" and "winner" are replaced with visually similar variants, like "fr3e" and "w1nner". This tactic is designed to confuse models that rely on exact keyword matching, making it more difficult for them to identify spam based solely on word frequencies.
2. **Zero-Width Character Insertion:** This attack involves inserting invisible zero-width spaces between characters of spam keywords, such as "fr[ZWSP]ee" or "w[ZWSP]inner". The goal is to deceive tokenizers, which may fail to recognize the

original words due to the inserted invisible characters. This technique is commonly used to bypass simple text-based classifiers.

3. **Synonym Replacement:** In this attack, common spam keywords are replaced with less frequent but semantically similar words. For example, the word “free” might be replaced with “complimentary”, or “winner” could be replaced with “champion”. While this does not alter the meaning of the message, it challenges models that rely on exact keyword matching and may affect their ability to classify the message correctly.

As shown in Table 4, the performance of the MNB model degrades sharply when handling these obfuscated messages. The naive Bayes classifier heavily relies on keyword frequencies, and any manipulation of the words (such as through spelling obfuscation or synonym replacement) significantly affects its ability to correctly classify the messages. For example, in the case of spelling obfuscation, the F1-score drops from 94.6% for standard spam to just 65.2%. Similarly, the MNB model’s performance on zero-width characters and synonym replacement is also significantly impacted, as the obfuscated words do not match the keywords it relies on.

In contrast, the DeBERTa model, which uses contextual understanding rather than simple keyword matching, demonstrates much better robustness against these attacks. The model is able to handle spelling obfuscation, zero-width characters, and synonym replacement with minimal performance degradation, maintaining F1-scores of 92.1%, 93.5%, and 92.8%, respectively. This resilience is due to DeBERTa’s ability to capture deeper semantic and contextual relationships between words, allowing it to recognize spam even when the message’s wording is altered.

Our hybrid model, while experiencing some performance drop, shows a significantly smaller decline than the pure MNB model and outperforms the pure DeBERTa model in terms of F1-score. For instance, the hybrid model achieves an F1-score of 92.5% on spelling obfuscation, 94.0% on zero-width characters, and 93.1% on synonym replacement. This indicates that the DeBERTa branch greatly enhances the model’s resilience to semantic variations, while the MNB branch continues to work efficiently on unaffected samples, leading to superior overall robustness.

In summary, the hybrid model combines the strengths of both naive Bayes and DeBERTa, ensuring high robustness against various adversarial attacks while maintaining performance on standard spam detection tasks. The results underscore the effectiveness of the hybrid approach in handling diverse types of spam manipulation, making it a highly reliable solution for real-world spam filtering applications.

Table 4. Performance on adversarial samples (F1-score).

Model	Standard Spam	Spelling Obfuscation	Zero-Width Chars	Synonym Replacement
MNB (TF-IDF)	94.6%	65.2%	71.5%	78.4%
DeBERTa	95.7%	92.1%	93.5%	92.8%
Hybrid (Ours)	96.5%	92.5%	94.0%	93.1%

4.8. Error Analysis

To gain insight into the limitations of our hybrid model, we conducted a brief analysis of the misclassified samples (Table 5). We observed that false positives were often triggered by benign messages containing words frequently associated with spam, such as “free” or “win”, leading to overconfident predictions by the naive Bayes component. On the other hand, false negatives were typically short or obfuscated spam messages that lacked strong lexical cues, posing challenges even for the contextual understanding of DeBERTa. These observations highlight the need for improved handling of ambiguous and adversarial inputs, which we consider as directions for future work.

Table 5. Average inference time per sample for different models.

Model	Avg. Inference Time (ms)
Naive Bayes (TF-IDF)	0.45
DistilBERT	13.2
RoBERTa-base	29.5
DeBERTa-base	27.8
Hybrid (Ours)	28.0

5. Ethical Considerations

In developing and evaluating our spam detection framework, we acknowledge several ethical dimensions that warrant attention for responsible research and real-world deployment.

5.1. User Privacy

Our model is trained and evaluated on the publicly available SMS Spam Collection dataset, which contains anonymized text messages without personally identifiable information. However, in practical applications, spam detection systems are often applied to private communication content. Therefore, privacy-preserving measures such as on-device inference, data anonymization, and compliance with regulations like GDPR should be considered before deployment.

5.2. Bias and Fairness

The dataset used in this study exhibits a class imbalance, with a significantly higher proportion of ham messages compared to spam. This imbalance may lead to biased predictions if not properly addressed. While we have applied class weighting and evaluated performance using metrics suitable for imbalanced data, further steps such as fairness audits and bias mitigation strategies are recommended, especially when the system is extended to diverse linguistic or cultural settings.

5.3. Responsible Deployment

Spam detection systems can influence message deliverability and user experience. Overly aggressive filters may lead to legitimate messages being blocked (false positives), while overly permissive systems may allow harmful content to pass through (false negatives). We recommend that such systems be deployed with human-in-the-loop mechanisms, configurable thresholds, and ongoing monitoring. Additionally, transparency through interpretability techniques can enhance trust and accountability in automated decisions.

6. Conclusions

In this paper, we presented a hybrid spam detection framework that effectively combines a traditional naive Bayes classifier and a large language model, DeBERTa, through a simple yet effective weighted probability fusion mechanism. Our approach achieves a favorable balance between computational efficiency and classification accuracy. Experimental results on the SMS Spam Collection dataset demonstrate that the proposed hybrid model consistently outperforms both individual baselines, achieving up to 97.5% accuracy and 96.5% F1-score. This demonstrates the potential of combining classical statistical models and LLMs for practical, high-performance spam filtering.

In future work, we plan to extend this approach to other security-related text classification tasks such as phishing detection and toxic comment filtering. Additionally, we aim to explore more adaptive fusion strategies and investigate the effectiveness of

lightweight LLM variants to further optimize the trade-off between performance and computational cost.

Author Contributions: Conceptualization, A.Z. and H.W.; methodology, A.Z.; software, A.Z.; validation, K.L., A.Z. and H.W.; formal analysis, A.Z.; investigation, K.L.; resources, A.Z.; data curation, K.L.; writing—original draft preparation, A.Z.; writing—review and editing, K.L. and H.W.; visualization, K.L.; supervision, H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

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

References

1. Karim, A.; Azam, S.; Shanmugam, B.; Kannoorpatti, K.; Alazab, M. A comprehensive survey for intelligent spam email detection. *IEEE Access* **2019**, *7*, 168261–168295. [\[CrossRef\]](#)
2. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Abdulhamid, S.M.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon* **2019**, *5*, e01802. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Xia, W.; Peng, R.; Chu, H.; Zhu, X.; Yang, Z.; Yang, L.; Lv, B.; Xiang, X. An Improved Pure Fully Connected Neural Network for Rice Grain Classification. *arXiv* **2025**, arXiv:2503.03111.
4. Zhao, C.-T.; Wang, R.-F.; Tu, Y.-H.; Pang, X.-X.; Su, W.-H. Automatic lettuce weed detection and classification based on optimized convolutional neural networks for robotic weed control. *Agronomy* **2024**, *14*, 2838. [\[CrossRef\]](#)
5. Wang, R.F.; Su, W.H. The application of deep learning in the whole potato production Chain: A Comprehensive review. *Agriculture* **2024**, *14*, 1225. [\[CrossRef\]](#)
6. Mishra, J.; Sahay, S.K. Modern Hardware Security: A Review of Attacks and Countermeasures. *arXiv* **2025**, arXiv:2501.04394. [\[CrossRef\]](#)
7. Samarthrao, K.V.; Rohokale, V.M. Enhancement of email spam detection using improved deep learning algorithms for cyber security. *J. Comput. Secur.* **2022**, *30*, 231–264. [\[CrossRef\]](#)
8. Altulaihan, E.; Alismail, A.; Hafizur Rahman, M.; Ibrahim, A.A. Email security issues, tools, and techniques used in investigation. *Sustainability* **2023**, *15*, 10612. [\[CrossRef\]](#)
9. Joung, J.; Kim, H.M. Automated keyword filtering in latent Dirichlet allocation for identifying product attributes from online reviews. *J. Mech. Des.* **2021**, *143*, 084501. [\[CrossRef\]](#)
10. Boger, Z.; Kuflik, T.; Shoval, P.; Shapira, B. Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems. *Inf. Process. Manag.* **2001**, *37*, 187–198. [\[CrossRef\]](#)
11. Sahami, M.; Dumais, S.; Heckerman, D.; Horvitz, E. A Bayesian approach to filtering junk e-mail. In Proceedings of the Learning for Text Categorization: Papers from the 1998 Workshop, Madison, WI, USA, 26–27 July 1998; Volume 62, pp. 98–105.
12. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [\[CrossRef\]](#)
13. Azam, N.; Yao, J. Comparison of term frequency and document frequency based feature selection metrics in text categorization. *Expert Syst. Appl.* **2012**, *39*, 4760–4768. [\[CrossRef\]](#)
14. Hakim, A.A.; Erwin, A.; Eng, K.I.; Galinium, M.; Muliady, W. Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In Proceedings of the 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 7–8 October 2014; pp. 1–4.
15. Zhang, C.; Jiang, H.; Liu, W.; Li, J.; Tang, S.; Juhas, M.; Zhang, Y. Correction of out-of-focus microscopic images by deep learning. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 1957–1966. [\[CrossRef\]](#)
16. Zheng, F.; Chen, X.; Liu, W.; Li, H.; Lei, Y.; He, J.; Pun, C.M.; Zhou, S. SMAFormer: Synergistic Multi-Attention Transformer for Medical Image Segmentation. In Proceedings of the 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Lisbon, Portugal, 3–6 December 2024.
17. Liu, W.; Shen, X.; Pun, C.M.; Cun, X. Explicit visual prompting for low-level structure segmentations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 19434–19445.

18. Liu, W.; Cun, X.; Pun, C.M.; Xia, M.; Zhang, Y.; Wang, J. Coordfill: Efficient high-resolution image inpainting via parameterized coordinate querying. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 1746–1754.
19. Liu, W.; Shen, X.; Li, H.; Bi, X.; Liu, B.; Pun, C.M.; Cun, X. Depth-aware Test-Time Training for Zero-shot Video Object Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 19218–19227.
20. Liu, W.; Juhas, M.; Zhang, Y. Fine-grained breast cancer classification with bilinear convolutional neural networks (BCNNs). *Front. Genet.* **2020**, *11*, 547327. [[CrossRef](#)]
21. Jiang, H.; Li, S.; Liu, W.; Zheng, H.; Liu, J.; Zhang, Y. Geometry-aware cell detection with deep learning. *Msystems* **2020**, *5*, 10–1128. [[CrossRef](#)]
22. Liu, W.; Qian, J.; Yao, Z.; Jiao, X.; Pan, J. Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection. *Future Internet* **2019**, *11*, 115. [[CrossRef](#)]
23. Liu, W.; Cun, X.; Pun, C.M. DH-GAN: Image manipulation localization via a dual homology-aware generative adversarial network. *Pattern Recognit.* **2024**, *155*, 110658. [[CrossRef](#)]
24. Liu, W.; Shen, X.; Pun, C.M.; Cun, X. ForgeryTTT: Zero-Shot Image Manipulation Localization with Test-Time Training. *arXiv* **2024**, arXiv:2410.04032.
25. Zhu, L.; Liu, W.; Chen, X.; Li, Z.; Chen, X.; Wang, Z.; Pun, C.M. Test-Time Intensity Consistency Adaptation for Shadow Detection. *arXiv* **2024**, arXiv:2410.07695. [[CrossRef](#)]
26. Jiang, H.; Tang, S.; Liu, W.; Zhang, Y. Deep learning for COVID-19 chest CT (computed tomography) image analysis: A lesson from lung cancer. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 1391–1399. [[CrossRef](#)] [[PubMed](#)]
27. Li, X.; Huang, G.; Cheng, L.; Zhong, G.; Liu, W.; Chen, X.; Cai, M. Cross-domain visual prompting with spatial proximity knowledge distillation for histological image classification. *J. Biomed. Inform.* **2024**, *158*, 104728. [[CrossRef](#)]
28. Di, X.; Cui, K.; Wang, R.F. Toward Efficient UAV-Based Small Object Detection: A Lightweight Network with Enhanced Feature Fusion. *Remote Sens.* **2025**, *17*, 2235. [[CrossRef](#)]
29. Huo, Y.; Wang, R.F.; Zhao, C.T.; Hu, P.; Wang, H. Research on Obtaining Pepper Phenotypic Parameters Based on Improved YOLOX Algorithm. *AgriEngineering* **2025**, *7*, 209. [[CrossRef](#)]
30. Yang, Z.X.; Li, Y.; Wang, R.F.; Hu, P.; Su, W.H. Deep Learning in Multimodal Fusion for Sustainable Plant Care: A Comprehensive Review. *Sustainability* **2025**, *17*, 5255. [[CrossRef](#)]
31. Huang, J.; Feng, X.; Chen, Q.; Zhao, H.; Cheng, Z.; Bai, J.; Zhou, J.; Li, M.; Qin, L. MLDebugging: Towards Benchmarking Code Debugging Across Multi-Library Scenarios. *arXiv* **2025**, arXiv:2506.13824. [[CrossRef](#)]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
33. Li, H.; Pun, C.M. Cee-net: Complementary end-to-end network for 3d human pose generation and estimation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 1305–1313.
34. Li, H.; Pun, C.M. Monocular robust 3d human localization by global and body-parts depth awareness. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7692–7705. [[CrossRef](#)]
35. Li, H.; Ge, S.; Gao, C.; Gao, H. Few-shot object detection via high-and-low resolution representation. *Comput. Electr. Eng.* **2022**, *104*, 108438. [[CrossRef](#)]
36. Li, H.; Zheng, F.; Liu, Y.; Xiong, J.; Zhang, W.; Hu, H.; Gao, H. Adaptive Skeleton Prompt Tuning for Cross-Dataset 3D Human Pose Estimation. In Proceedings of the ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, 6–11 April 2025; pp. 1–5.
37. Li, H.; Pun, C.M.; Xu, F.; Pan, L.; Zong, R.; Gao, H.; Lu, H. A hybrid feature selection algorithm based on a discrete artificial bee colony for Parkinson’s diagnosis. *ACM Trans. Internet Technol.* **2021**, *21*, 63. [[CrossRef](#)]
38. Yang, S.; Li, H.; Pun, C.M.; Du, C.; Gao, H. Adaptive spatial-temporal graph-mixer for human motion prediction. *IEEE Signal Process. Lett.* **2024**, *31*, 1244–1248. [[CrossRef](#)]
39. Yan, X.; Xie, J.; Liu, M.; Li, H.; Gao, H. Hierarchical local temporal network for 2d-to-3d human pose estimation. *IEEE Internet Things J.* **2024**, *12*, 869–880. [[CrossRef](#)]
40. Wu, A.Q.; Li, K.L.; Song, Z.Y.; Lou, X.; Hu, P.; Yang, W.; Wang, R.F. Deep Learning for Sustainable Aquaculture: Opportunities and Challenges. *Sustainability* **2025**, *17*, 5084. [[CrossRef](#)]
41. Yang, Z.Y.; Xia, W.K.; Chu, H.Q.; Su, W.H.; Wang, R.F.; Wang, H. A comprehensive review of deep learning applications in cotton industry: From field monitoring to smart processing. *Plants* **2025**, *14*, 1481. [[CrossRef](#)]
42. Wang, Z.; Zhang, H.W.; Dai, Y.Q.; Cui, K.; Wang, H.; Chee, P.W.; Wang, R.F. Resource-Efficient Cotton Network: A Lightweight Deep Learning Framework for Cotton Disease and Pest Classification. *Plants* **2025**, *14*, 2082. [[CrossRef](#)] [[PubMed](#)]

43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (long and short papers), pp. 4171–4186. Available online: https://aclanthology.org/N19-1423/?utm_campaign=The+Batch&utm_source=hs_email&utm_medium=email&_hsenc=p2ANqtz-_m9bbH_7ECE1h3lZ3D61TYg52rKpifVNjL4fvJ85uqggrXsWDBTB7YooFLJeNXHWqhvOyC (accessed on 13 August 2025).
44. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
45. He, P.; Gao, J.; Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv* **2021**, arXiv:2111.09543.
46. Bai, J.; Yin, Y.; Dong, Y.; Zhang, X.; Pun, C.M.; Chen, X. LensNet: An End-to-End Learning Framework for Empirical Point Spread Function Modeling and Lensless Imaging Reconstruction. *arXiv* **2025**, arXiv:2505.01755.
47. Bai, J.; Yin, Y.; He, Q.; Li, Y.; Zhang, X. Retinexmamba: Retinex-based mamba for low-light image enhancement. In Proceedings of the International Conference on Neural Information Processing, Okinawa, Japan, 20–24 November 2025; pp. 427–442.
48. Chen, X.; Ng, M.K.P.; Tsang, K.F.; Pun, C.M.; Wang, S. ConnectomeDiffuser: Generative AI Enables Brain Network Construction from Diffusion Tensor Imaging. *IEEE Trans. Consum. Electron.* **2025**. [\[CrossRef\]](#)
49. Chen, X.; Li, Z.; Shen, Y.; Mahmud, M.; Pham, H.; Pun, C.M.; Wang, S. High-Fidelity Functional Ultrasound Reconstruction via A Visual Auto-Regressive Framework. *arXiv* **2025**, arXiv:2505.21530.
50. Li, M.; Sun, H.; Lei, Y.; Zhang, X.; Dong, Y.; Zhou, Y.; Li, Z.; Chen, X. High-Fidelity Document Stain Removal via A Large-Scale Real-World Dataset and A Memory-Augmented Transformer. In Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, 26 February–6 March 2025; pp. 7614–7624.
51. Zhou, Z.; Lei, Y.; Chen, X.; Luo, S.; Zhang, W.; Pun, C.M.; Wang, Z. DocDeshadower: Frequency-Aware Transformer for Document Shadow Removal. In Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kuching, Malaysia, 6–10 October 2024; pp. 2468–2473.
52. Guo, X.; Luo, S.; Dong, Y.; Liang, Z.; Li, Z.; Zhang, X.; Chen, X. An asymmetric calibrated transformer network for underwater image restoration. *Vis. Comput.* **2025**, *41*, 6465–6477. [\[CrossRef\]](#)
53. Guo, X.; Dong, Y.; Chen, X.; Chen, W.; Li, Z.; Zheng, F.; Pun, C.M. Underwater image restoration via polymorphic large kernel cnns. In Proceedings of the ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, 6–11 April 2025; pp. 1–5.
54. Guo, X.; Chen, X.; Wang, S.; Pun, C.M. Underwater Image Restoration Through a Prior Guided Hybrid Sense Approach and Extensive Benchmark Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 4784–4800. [\[CrossRef\]](#)
55. Yi, F.; Zheng, Z.; Liang, Z.; Dong, Y.; Fang, X.; Wu, W.; Chen, X. MAC-Lookup: Multi-Axis Conditional Lookup Model for Underwater Image Enhancement. *arXiv* **2025**, arXiv:2507.02270.
56. Zhang, X.; Chen, F.; Wang, C.; Tao, M.; Jiang, G.P. Sienet: Siamese expansion network for image extrapolation. *IEEE Signal Process. Lett.* **2020**, *27*, 1590–1594. [\[CrossRef\]](#)
57. Zhang, X.; Zhao, Y.; Gu, C.; Lu, C.; Zhu, S. SpA-Former: An Effective and lightweight Transformer for image shadow removal. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–8.
58. Xu, Z.; Zhang, X.; Chen, W.; Liu, J.; Xu, T.; Wang, Z. MuralDiff: Diffusion for Ancient Murals Restoration on Large-Scale Pre-Training. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2169–2181. [\[CrossRef\]](#)
59. Zhang, X.; Xu, Z.; Tang, H.; Gu, C.; Zhu, S.; Guan, X. *Shadclips: When Parameter-Efficient Fine-Tuning with Multimodal Meets Shadow Removal*; Research Square: Durham, NC, USA, 2024. [\[CrossRef\]](#)
60. Zhang, X.; Shen, C.; Yuan, X.; Yan, S.; Xie, L.; Wang, W.; Gu, C.; Tang, H.; Ye, J. From Redundancy to Relevance: Enhancing Explainability in Multimodal Large Language Models. *arXiv* **2024**, arXiv:2406.06579.
61. Wei, J.; Zhang, X. DOPRA: Decoding Over-accumulation Penalization and Re-allocation in Specific Weighting Layer. *arXiv* **2024**, arXiv:2407.15130.
62. Yuan, X.; Shen, C.; Yan, S.; Zhang, X.; Xie, L.; Wang, W.; Guan, R.; Wang, Y.; Ye, J. Instance-adaptive Zero-shot Chain-of-Thought Prompting. *arXiv* **2024**, arXiv:2409.20441.
63. Zhang, X.; Quan, Y.; Gu, C.; Shen, C.; Yuan, X.; Yan, S.; Cheng, H.; Wu, K.; Ye, J. Seeing Clearly by Layer Two: Enhancing Attention Heads to Alleviate Hallucination in LVLMS. *arXiv* **2024**, arXiv:2411.09968. [\[CrossRef\]](#)
64. Zhang, X.; Zeng, F.; Gu, C. Simignore: Exploring and enhancing multimodal large model complex reasoning via similarity computation. *Neural Netw.* **2024**, *184*, 107059. [\[CrossRef\]](#)
65. Zhang, X.; Zeng, F.; Quan, Y.; Hui, Z.; Yao, J. Enhancing Multimodal Large Language Models Complex Reason via Similarity Computation. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025.

66. Goodman, J.; Cormack, G.V.; Heckerman, D. Spam and the ongoing battle for the inbox. *Commun. ACM* **2007**, *50*, 24–33. [[CrossRef](#)]
67. Drucker, H.; Wu, D.; Vapnik, V.N. Support vector machines for spam categorization. *IEEE Trans. Neural Netw.* **1999**, *10*, 1048–1054. [[CrossRef](#)]
68. Metsis, V.; Androutsopoulos, I.; Paliouras, G. Spam filtering with naive bayes-which naive bayes? *CEAS* **2006**, *17*, 28–69.
69. Wang, R.F.; Qu, H.R.; Su, W.H. From Sensors to Insights: Technological Trends in Image-Based High-Throughput Plant Phenotyping. *Smart Agric. Technol.* **2025**, *12*, 101257. [[CrossRef](#)]
70. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
71. Zhao, Z. Using Pre-Trained Language Models for Toxic Comment Classification. Ph.D. Thesis, University of Sheffield, Sheffield, UK, 2022.
72. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.