Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.3, June 2025, pp. 592~602
e-ISSN: 2548-6861

Comparative Analysis of CNN, Transformers, and Traditional ML for Classifying Online Gambling Spam Comments in Indonesian

Martin Clinton Tosima Manullang [1*], Arkham Zahri Rakhman [2], Hartanto Tantriawan [3], Andika Setiawan [4]
[*] Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Sumatera, Lampung Selatan, Indonesia
martin.manullang@if.itera.ac.id [1*], arkham@if.itera.ac.id [2], hartanto.tantriawan@if.itera.ac.id [3], andika.setiawan@if.itera.ac.id [4]

Article Info

Article history:

Received 2025-04-22
Revised 2025-05-07
Accepted 2025-06-02

Keyword:

Indonesian Language,
Deep Learning,
Spam Detection,
Transformer,
Wordformer.

ABSTRACT

The rise of user-generated content on social media and live-streaming platforms has intensified the spread of spam, particularly online gambling (Judi Online) promotions, which remain prevalent in Indonesian comment sections. This study investigates the effectiveness of various machine learning (ML) and deep learning (DL) approaches in classifying such spam content in Bahasa Indonesia. We compare five models: Support Vector Machine (SVM), Random Forest (RF), a CNN-based model, IndoBERT, and a custom lightweight transformer model named Wordformer. While IndoBERT achieves the highest performance across all metrics, it comes with high computational demands. Wordformer, in contrast, delivers a strong balance between accuracy and efficiency, outperforming traditional models while being significantly more lightweight than IndoBERT. Wordformer achieved 0.9975 accuracy and macro F1-score, surpassing SVM (0.9578) and Random Forest (0.9729), while maintaining a significantly smaller model size and fewer multiply-add operations. An extensive ablation study further explores the architectural and training design choices that influence Wordformer's performance. The findings suggest that lightweight transformer models can offer practical, scalable solutions for spam detection in low-resource language settings without the need for large pretrained backbones.

I. INTRODUCTION

In the digital era, the rise of user-generated content on platforms like social media and online comment sections has enabled unprecedented public interaction—but also opened the floodgates to misuse, including the spread of spam promoting Judi Online (online gambling). In Indonesia, this issue has become more than a nuisance; it is now a serious societal and economic threat. Despite strict legal bans, online gambling remains widely accessible, especially to young people, and has been linked to addiction, mental health decline, and even suicidal behavior [1]. Economically, the consequences ripple beyond individuals—job loss, debt, asset liquidation, and criminal behavior to finance gambling are increasingly reported, leading to social instability and reduced workforce productivity [2]. This multifaceted harm underscores the urgency of effective spam filtering systems, especially those capable of understanding the linguistic patterns of Bahasa Indonesia.

Online gambling promotional comment spam has emerged as a concerning digital phenomenon in Indonesia, not only disrupting user experience on the internet but also exacerbating the broader social and economic consequences of online gambling practices. Sriyana, et al. [3] highlights that online gambling has severe psychological effects on individuals, including increased anxiety, social isolation, and family conflicts. The proliferation of spam comments has become more deeply rooted as public figures increasingly endorse illegal gambling sites, as noted by Kosasih and Setiady [4], who argue that such celebrity involvement distorts public perception and blurs the line between legitimate and illicit promotion. Additionally, Nurdiansyah and Kanda [5] emphasize the economic losses and mental health issues triggered by online gambling, further intensified by the accessibility of such content through spam. Rafiqah

and Rasyid [6] point out that these impacts also contribute to the degradation of social values within families and communities. Hadi et al. [7] underscore the importance of education in combating the spread of online gambling content, particularly spam comments frequently found on social media and public forums. Collectively, these studies reinforce the argument that spam comments promoting online gambling are not merely a nuisance but represent a complex societal threat that undermines the moral and social resilience of Indonesian society.

According to a review by Chrismanto et al. [8], detecting spam in Bahasa Indonesia is non-trivial due to its linguistic complexity. Informal grammar, regional slang, frequent abbreviations, and code-mixing with English make preprocessing—like tokenization and normalization—more difficult. These issues are worsened by the scarcity of standardized datasets for Bahasa Indonesia, limiting model development and comparison. Additionally, spam comments are often short, emoji-heavy, and context-dependent, making them harder to detect using typical keyword-based or language-agnostic approaches.

We review some approaches to text classification from a top-cited paper [9]. Early approaches to text classification largely relied on traditional machine learning (ML) models such as Support Vector Machines (SVM) [10] and Random Forests (RF) [11], which depend heavily on manual feature engineering—like bag-of-words, TF-IDF, and POS-tagged terms. While effective to a degree, these models struggle with capturing contextual and sequential nuances in language. As tasks became more complex and data grew, deep learning (DL) models began to dominate. Convolutional Neural Networks (CNNs) proved effective in detecting local patterns like spam phrases, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks helped in modeling sequential dependencies. More recently, transformer-based architectures such as BERT [12] have redefined the state of the art by leveraging self-attention mechanisms to model global context without the sequential limitations of RNNs.

However, these advancements come with a trade-off: performance vs. efficiency. Large-scale models like BERT offer superior accuracy but are computationally expensive and memory-intensive. This has led to a growing interest in lightweight models—including compact CNNs and efficient transformers—that seek to balance strong performance with lower resource demands [13]. This study embraces that trend by evaluating both traditional ML and diverse DL models, including a custom lightweight transformer, for spam detection in Bahasa Indonesia.

In recent years, CNN-based models for NLP have seen renewed interest, particularly in scenarios where computational constraints or real-time requirements limit the feasibility of larger transformer architectures. Unlike their use in computer vision, CNNs in text classification act as local n-gram detectors, capturing short-range dependencies with relatively low computational cost. Variants like TextCNN [14], CharCNN [15], and dilated CNNs have shown strong performance on benchmark tasks such as sentiment analysis, spam detection, and question classification, especially when paired with pretrained embeddings. However, their limitation lies in capturing long-range dependencies and global context—something that transformers handle more naturally.

On the other hand, transformer-based architectures have become the backbone of modern NLP systems due to their superior ability to model context and capture semantic nuances. Beyond BERT, variants like DistilBERT [16], ALBERT [17], TinyBERT [18], and MobileBERT [19] have been proposed to reduce model size and inference time while retaining much of the performance of their larger counterparts. More recent innovations such as Linformer, Performer, and Longformer aim to reduce the quadratic complexity of self-attention, making transformers more scalable for longer sequences or deployment on low-resource devices. These models have demonstrated that with careful design, efficiency and effectiveness can co-exist, opening new opportunities for building real-time NLP applications even in low-resource settings.

Despite advances in content moderation, spam comments promoting Judi Online continue to flood Indonesian digital spaces—particularly during live YouTube sessions hosted by local streamers and content creators. The persistence of such content, even on platforms as technologically advanced as Google's YouTube, underscores the challenge of filtering spam in low-resource languages like Bahasa Indonesia.

This study is motivated by the urgent need for more effective, language-specific spam detection models. We aim to compare the performance of traditional machine learning models (SVM, RF), standard deep learning approaches (CNN, BERT), and our custom lightweight transformer architecture called Wordformer. While models like BERT offer state-of-the-art accuracy, their computational demands limit real-time deployment on resource-constrained systems. Thus, our study seeks not only to measure predictive accuracy but also to highlight the trade-offs between efficiency and effectiveness—crucial for practical applications in regional and low-resource settings.

This study addresses the persistent challenge of detecting online gambling (Judi Online) spam comments in Bahasa Indonesia—a task complicated by informal language use and limited NLP resources. While existing solutions have shown promise in high-resource settings, their effectiveness in regional languages remains insufficient. The core problem lies in identifying which modeling approach—traditional ML, standard DL, or lightweight architectures—best balances accuracy and computational efficiency. To this end, our study poses two primary research questions:

1) How do traditional machine learning models and deep learning architectures perform in classifying Indonesian online gambling spam comments?
2) Can Wordformer, our proposed lightweight transformer model, deliver competitive performance while significantly reducing computational overhead?

The primary objective of this study is to evaluate and compare the performance of various machine learning and deep learning models—including SVM, Random Forest, CNN, BERT, and our custom Wordformer—for classifying Judi Online spam comments written in Bahasa Indonesia. The evaluation considers both predictive performance and computational efficiency. The scope is limited to monolingual spam detection using a custom-labeled dataset collected from Indonesian YouTube comment sections. While basic text preprocessing is applied, the study does not incorporate advanced linguistic techniques such as syntactic parsing or cross-lingual modeling. Additionally, the focus remains on offline classification performance, without integrating the models into real-time systems or deploying them in production environments.

This study contributes to the growing body of research on spam detection by offering a focused evaluation of model performance in a low-resource, language-specific context. By comparing a range of traditional and modern architectures—including the proposed Wordformer—it provides valuable insight into how accuracy and computational cost trade off in real-world spam filtering scenarios. The inclusion of a lightweight yet high-performing transformer model highlights the feasibility of deploying efficient solutions on regional platforms, especially where computational resources are limited. Ultimately, this work supports the development of more robust moderation tools tailored to the linguistic and behavioral characteristics of Indonesian online communities.

This thesis is structured as follows. Chapter 2 presents a literature review covering prior research on spam detection and relevant machine learning and deep learning techniques. Chapter 2 details the methodology, including dataset construction, preprocessing steps, and model configurations. Chapter 3 reports and compares the results. Chapter 4 discusses the findings, limitations, and implications. Finally, Chapter 5 concludes the study and suggests directions for future work.

II. METHODS

This chapter outlines the methodology employed to investigate the performance of various machine learning and deep learning models for classifying online gambling (Judi Online) spam comments in Bahasa Indonesia. The focus is on ensuring a fair and consistent evaluation across all approaches by using a standardized dataset, uniform preprocessing pipeline, and shared evaluation protocol. We begin by describing the dataset and preprocessing steps, followed by a breakdown of the model architectures—including traditional classifiers, deep learning baselines, and our proposed lightweight transformer model, Wordformer. The chapter concludes with details on training configurations, performance metrics, and the computing environment used for experimentation.

A. Dataset

This study utilizes a publicly available dataset titled "Deteksi Judi Online" sourced from Kaggle (https://www.kaggle.com/datasets/yaemico/deteksi-judi-online) [20]. The dataset comprises 6,351 user comments collected from the live YouTube broadcast "Wayang Jogja Night Carnival #9", hosted by the @TribunJogjaOfficial channel in celebration of the 268th anniversary of Yogyakarta City. The data was originally gathered to analyze the presence of online gambling (Judi Online) promotions in live chat interactions.

Each data entry includes metadata such as the timestamp (datetime), commenter name (author_name), the original comment (message), a pre-cleaned version (cleaned_message), and a binary label indicating whether the comment contains online gambling promotion (1) or not (0). For the purpose of this research, we chose to ignore the cleaned_message field and instead applied our own standardized preprocessing pipeline to ensure consistency across all models.

All experiments were conducted using 5-fold cross-validation, with the same data splits applied uniformly across every evaluated method to ensure fair comparison and reduce variance due to data partitioning.

The dataset was annotated manually by the original dataset creator and includes binary labels (1 for online gambling spam, 0 for non-spam). Since it is a public dataset, it supports reproducibility and provides a useful benchmark for future studies. While the dataset is specific to a single event and YouTube source, it reflects real-world usage and typical spam behavior in Indonesian live chat environments.

B. Data Preprocessing

To prepare the input text for modeling, a standardized preprocessing pipeline was applied to the raw message field. Each step was designed to clean informal Indonesian comments while preserving the semantic content relevant to spam classification. The following transformations were used:

- Lowercasing: All text was converted to lowercase to reduce vocabulary size. Example: "GRATIS JOIN SEKARANG!!!" become "gratis join sekarang!!!"
- URL removal: Spam links were removed using regular expressions. Example: "kunjungi http://judi123.com sekarang!" become "kunjungi sekarang!"
- Mention and hashtag removal: Mentions (@username) and hashtags (#promo) were stripped as they added little value. Example: "@user ayo join sekarang #judi" become "ayo join sekarang"
- Emoji and symbol cleaning: Non-textual characters such as emojis and decorative symbols were replaced with spaces. Example: "🎰 join sekarang !!" become "join sekarang"
- Whitespace normalization: Extra spaces were collapsed to single spaces for token consistency.

After preprocessing, tokenization was model-dependent. For BERT-based models, we used the tokenizer from the pretrained model indobenchmark/indobert-base-p1, a language model specifically trained on large-scale Bahasa Indonesia corpora. This tokenizer applies subword segmentation using the WordPiece algorithm, which helps handle out-of-vocabulary and rare Indonesian terms.

For other models (CNN, Wordformer, and traditional ML), we used a simple whitespace-based tokenizer to split preprocessed text into tokens. This consistent pipeline ensured that all models received comparable inputs, while the use of a domain-appropriate tokenizer (IndoBERT) preserved language-specific features crucial for high-quality representation learning in Bahasa Indonesia.

The preprocessing steps applied in this study include lowercasing, URL removal, punctuation and symbol cleaning, and stopword removal using the NLTK Indonesian stopword list. Tokenization is performed using nltk.word_tokenize, which handles basic segmentation of informal Indonesian text. While this study does not explicitly normalize disguised spam variants, the models are trained on real spam patterns, enabling them to implicitly learn such variations. Addressing spelling variations and slang normalization remains a potential direction for future work.

C. Data Augmentation

To improve model robustness and reduce overfitting, we applied text augmentation techniques during training. Augmentation was controlled via a probability parameter p_augment (default: 0.5), meaning each training sample had a 50% chance of being augmented. The augmentation framework supported various methods designed to simulate lexical variation and noise commonly found in spam content, especially in informal Indonesian.

The methods include synonym replacement, where random non-stopword tokens are substituted with synonyms using WordNet; random deletion, which removes tokens with a small probability to mimic noisy inputs; random swap, where two words in the sentence are randomly exchanged; and random insertion, where new synonyms are inserted at random positions. Additionally, a simulated back-translation mechanism introduces minor distortions such as article removal, word reordering, and synonym substitution to mimic translation artifacts. A cascaded augmentation mode sequentially applies multiple of these techniques for greater diversity.

When the augmentation method is set to 'random', one of these strategies is selected randomly per sample. The augmentation system supports Bahasa Indonesia by filtering stopwords based on language-specific lists, although synonym lookup remains a challenge due to limited Indonesian lexical resources. This strategy helps expose the models to lexical variability, improving generalization across informal or adversarially structured spam text.

D. Model Architectures

This section outlines the architectures and configurations of all models evaluated in this study, beginning with traditional machine learning approaches.

1). Traditional Machine Learning Models: To establish a baseline, two well-known traditional machine learning classifiers were employed: SVM and RF

SVM is a discriminative classifier that constructs a hyperplane to separate classes with the maximum margin. It is known for its robustness in high-dimensional spaces and its effectiveness in sparse text data. A linear kernel was used in this study to reduce computational complexity.

RF is an ensemble learning method that builds multiple decision trees during training and outputs the majority class as the final prediction. Its inherent capability to handle noisy data and perform implicit feature selection makes it suitable for this task.

Both models rely on TF-IDF (Term Frequency–Inverse Document Frequency) vectorization to transform raw text into numerical features. The text was first tokenized using whitespace-based splitting (after preprocessing), and n-grams ranging from unigrams to trigrams were extracted to capture both individual words and short phrases. The final feature vectors are sparse and high-dimensional, capturing frequency-weighted word and phrase occurrences. These are then fed into the respective classifiers without dimensionality reduction, allowing the models to leverage the full token structure.

For traditional machine learning models such as SVM and Random Forest, TF-IDF vectorization is used as the feature representation. For deep learning models like CNN and Wordformer, embeddings are randomly initialized and learned during training. This ensures that deep learning models start without any pretrained information, maintaining a fair experimental setup.

2). CNN based model: The CNN model used in this study is a lightweight, modular architecture designed specifically for short-text classification, inspired by Kim's CNN model [21]. Its structure follows a straightforward flow, with clearly defined stages in the forward pass that transform raw token sequences into a final prediction score.

The process begins with the embedding layer, which converts each token index into a fixed-size dense vector. This layer is implemented using the nn.Embedding in the PyTorch framework, where the embeddings are randomly initialized and trained jointly with the model parameters. This approach is consistently applied to both the CNN and Wordformer models to ensure a fair comparison between architectures.

The CNN model processes tokenized text through parallel 1D convolutional layers with kernel sizes of 3, 4, and 5, each acting as n-gram detectors. Max-over-time pooling is applied to each feature map, selecting the most important activation per channel. The pooled outputs are concatenated, passed through a dropout layer, and finally classified by a fully connected linear layer. This multi-channel 1D CNN structure

enables the model to capture diverse local patterns within short Indonesian spam comments.

The embedding dimension is set to 300, and the embedding layer is trainable, allowing it to adapt to domain-specific usage common in Indonesian spam text. The output of this layer is a tensor of shape (`batch_size, sequence_length, embedding_dim`), which is then permuted to align with PyTorch's expected input for convolutional operations, resulting in a shape of (`batch_size, embedding_dim, sequence_length`).

Next, the model applies three parallel 1D convolutional layers, each with different kernel sizes—specifically 3, 4, and 5. These layers act as n-gram feature extractors, where the kernel size determines how many consecutive tokens are considered at a time. Each convolutional block uses 100 output channels, meaning that for each kernel size, the model learns 100 distinct filters. A ReLU activation is applied immediately after each convolution to introduce non-linearity.

Following convolution, each feature map undergoes max pooling over time. This operation compresses each feature map to its most significant activation, resulting in one value per filter. Since there are 100 filters for each of the three kernel sizes, the pooled outputs produce a combined feature vector of size 300 per input comment (100 × 3). These outputs are concatenated and passed through a dropout layer with a dropout rate of 0.5 to prevent overfitting.

Finally, the feature vector is processed by a fully connected linear layer that reduces the 300-dimensional vector to a single logit. This output is passed through a sigmoid activation function to produce a probability between 0 and 1, indicating the likelihood that the input comment is promoting online gambling. The visualisation of this deep learning architecture can be seen on Figure 1.

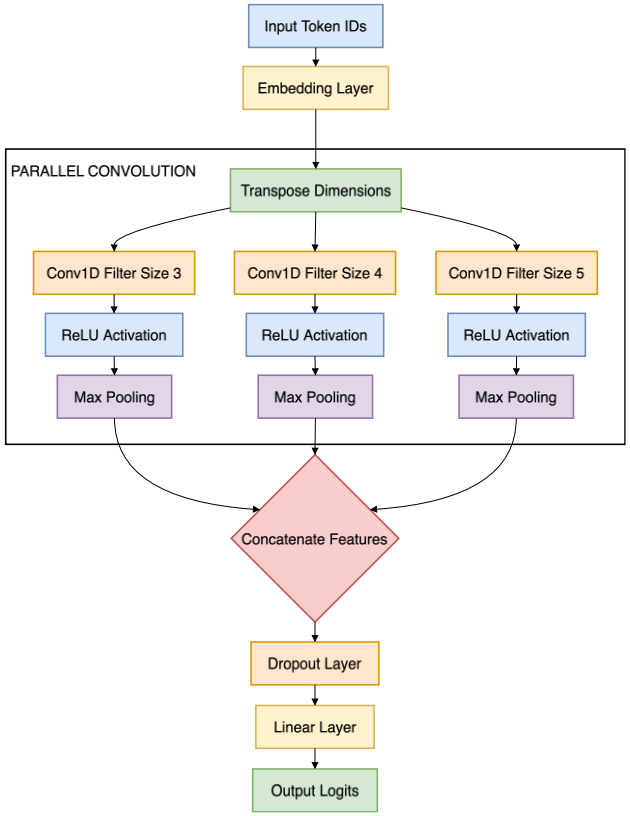

Figure 1. The CNN based architecture

3). BERT: Bidirectional Encoder Representations from Transformers (BERT) is a deep learning model developed by Google that has become a cornerstone in modern NLP. Unlike traditional models that read text sequentially, BERT uses a bidirectional transformer architecture to capture both left and right context simultaneously. This allows BERT to better understand the meaning of words based on surrounding context, which is especially useful for ambiguous or informal expressions common in social media comments.

In this study, we use the indobenchmark/indobert-base-p1 variant [22], a version of BERT pretrained specifically on large-scale Bahasa Indonesia corpora. This choice ensures the model is well-adapted to the linguistic characteristics of Indonesian text, providing an advantage over general multilingual models. The model can be obtained from the internet using this link https://huggingface.co/indobenchmark/indobert-base-p1. The fine-tuning process involves adding a simple classification head (a fully connected linear layer) on top of the [CLS] token output from BERT. The [CLS] token serves as a summary representation of the entire input sequence. During training, the entire BERT model, including the pretrained transformer layers is updated.

IndoBERT shares the same architecture as the original BERT-Base model proposed by Devlin et al. [12]: it consists of 12 transformer layers (encoder blocks), each with 12 self-attention heads, a hidden size of 768, and a feedforward layer of size 3072. The model contains approximately 124.5 million

parameters and uses SentencePiece with Byte-Pair Encoding (BPE) to tokenize inputs into subwords, allowing it to manage morphological variants commonly found in Bahasa Indonesia. BERT architecture can be seen on Figure 2.

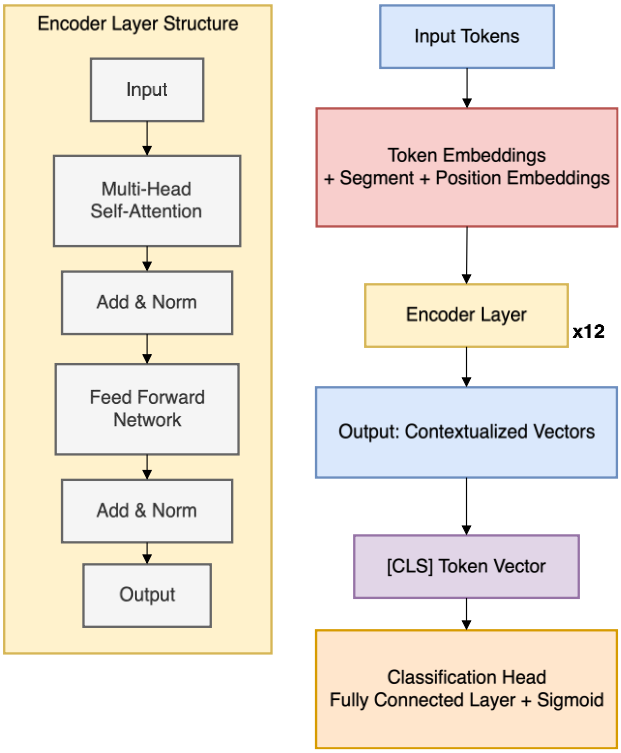

Figure 2. BERT architecture

4). Wordformer: To address the need for an efficient yet expressive spam classification model, this study introduces Wordformer, a lightweight transformer-based architecture designed from scratch. Notably, unlike TinyBERT and DistilBERT which employ knowledge distillation from pre-trained BERT models, Wordformer is trained entirely from scratch with randomly initialized weights and does not rely on any pre-trained BERT parameters. The model draws inspiration from transformer encoders like those used in BERT but is significantly simplified for computational efficiency. The input token IDs are first converted into dense vectors through a trainable embedding layer of shape (`vocab_size, emb_dim`), where the embedding dimension is set to 128. To preserve sequential information, which transformers do not inherently capture a sinusoidal positional encoding is added to the embeddings. The core of Wordformer is a single nn.TransformerEncoderLayer module, which processes the input in parallel using multi-head self-attention. This layer is configured with a model dimension of 128, 4 attention heads, a 256-dimensional feedforward layer, ReLU activation, and a dropout rate of 0.1. The output from this encoder layer is passed through a nn.TransformerEncoder, capturing contextual representations across the sequence.

Unlike BERT, Wordformer does not rely on a special [CLS] token; instead, it applies mean pooling across the token dimension to produce a single vector that represents the entire input sequence. This pooled output is regularized using dropout and then passed to a final linear layer with a single output unit, followed by a sigmoid activation to generate a binary prediction. The forward method cleanly outlines this sequence: embedding, positional encoding, transformer encoding, mean pooling, and classification. This compact design allows Wordformer to offer much lower memory and runtime cost compared to BERT, while still benefiting from the strength of attention-based representation learning. As such, Wordformer is well-suited for real-time spam moderation scenarios, especially in environments with limited computational resources. The detailed architecture of Wordformer can be seen on Figure 3.

Technically, Wordformer consists of 3 stacked transformer encoder layers, each using multi-head self-attention with 4 attention heads, and a 256-dimensional feed-forward network. The embedding size is set to 128, and a sinusoidal positional encoding is added to preserve token order information. The model includes LayerNorm, dropout (rate 0.1), and GELU activation in the feedforward sublayer. Token representations are pooled using mean pooling (instead of a [CLS] token), and passed through a linear classifier. This entire architecture is implemented using nn.TransformerEncoderLayer in PyTorch and designed to remain light and interpretable

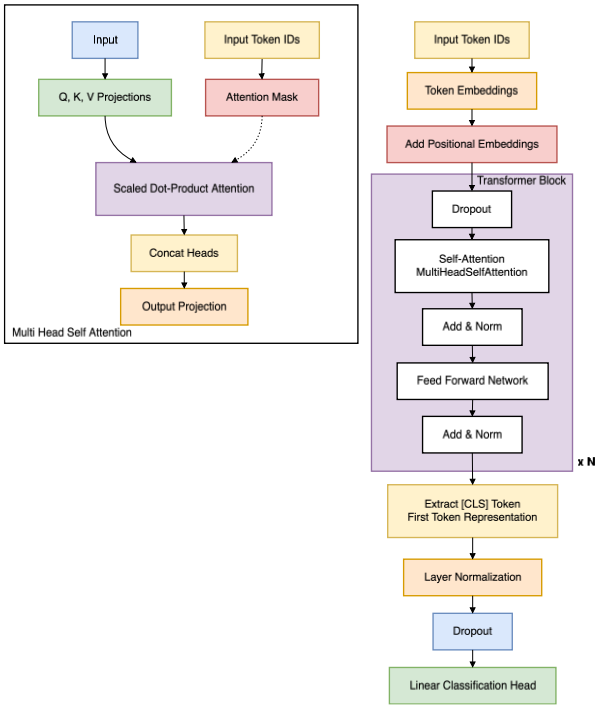

Figure 3. Wordformer Architecture

E. Training Strategy

All models in this study were trained using a unified and reproducible strategy designed for robust evaluation and fair comparison. For deep learning models (CNN, BERT, and Wordformer), training was performed using binary cross-entropy loss, optimized with the Adam optimizer. The initial learning rate was set to 1e-5, with weight decay of 0.01 applied for regularization. A learning rate scheduler with `ReduceLROnPlateau` strategy was used: if the validation score did not improve after 2 epochs (`lr_patience`), the learning rate was reduced by a factor of 0.5. This helped prevent stagnation during optimization.

Training was conducted for a maximum of 35 epochs with a batch size of 32, and a maximum sequence length of 128 for all models. Early stopping was enabled, with a patience of 6 epochs, meaning training would halt if the validation F1-score did not improve within that window. During training, the best-performing model checkpoint (based on validation F1-score) was saved for each fold.

To improve generalization and mitigate overfitting, dropout was applied (with a rate of 0.2) before the classification layers. Additionally, five-fold cross-validation was used across all models, ensuring that the same data splits were consistently applied.

For reproducibility, the random seed was fixed to 42, and all experiments were logged using Weights & Biases (wandb) when enabled. Model outputs, logs, and checkpoints were stored in a specified output directory for each run. This setup ensures not only fair and consistent evaluation across different model families but also supports stable convergence and transparent model tracking throughout the training lifecycle.

F. Performance Metrices

To comprehensively evaluate model performance, we employed a range of classification metrics computed on the validation set for each fold. These include accuracy, precision, recall, F1-score (macro and weighted), AUC (Area Under the ROC Curve), and the confusion matrix.

Accuracy measures the proportion of correct predictions across all classes:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Precision calculates the ratio of correctly predicted positive observations to all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (or sensitivity) calculates the ratio of correctly predicted positive observations to all actual positives:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score is the harmonic mean of precision and recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

We report both macro-F1, which averages F1 scores equally across all classes, and weighted-F1, which considers class imbalance by weighting each class by its support. AUC evaluates the ranking quality of predicted probabilities, capturing the model's ability to distinguish between classes. Validation Loss is calculated using binary cross-entropy, measuring the model's confidence in its probabilistic predictions. Confusion Matrix provides a tabular breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), giving a direct view of classification correctness per class.

Together, these metrics offer a balanced assessment of both general performance and the model's handling of class imbalance—critical in spam detection tasks where false positives and false negatives carry different practical risks.

G. Computing Environment

All experiments in this study were conducted using a high-performance workstation equipped with an Intel(R) Xeon(R) Silver 4215R CPU @ 3.20GHz, 32 GB of RAM, and an NVIDIA Quadro RTX 5000 GPU. The models were implemented using the PyTorch deep learning framework with CUDA support, enabling efficient training of resource-intensive architectures like BERT and Wordformer.

To ensure broader accessibility, the implementation is compatible with CPU-only environments and Apple's Metal Performance Shaders backend for macOS. However, it is important to note that while the code can run on these platforms, performance may be significantly slower and not suitable for large-scale training.

III. RESULTS

This chapter presents the experimental results obtained from evaluating five different models on the task of detecting online gambling spam comments in Bahasa Indonesia. The evaluation is divided into two main parts: a benchmark comparison of three deep learning models (CNN, IndoBERT, and the proposed Wordformer) and two traditional machine learning algorithms (SVM and Random Forest), followed by an ablation study examining the impact of key architectural and training hyperparameters on Wordformer's performance. All results are reported using standard classification metrics and are averaged across five-fold cross-validation to ensure robustness and reliability. All running results were logged into Wandb platform and can be accessed using this url https://wandb.ai/mctosima/youtube-comment-spam. We intentionally open the code to offer more opportunity for reproducing this study in https://github.com/mctosima/online_bet_spam_detection

TABLE I
BENCHMARK RESULTS

Method	Accuracy	Weighted F1	Macro F1	Precision	Recall	AUC
Support Vector Machine	0.957781	0.95772	0.9577	0.957781	0.9577819	0.9577
Random Forest	0.9729048	0.972893	0.972895	0.9522081	0.97299022	0.97289
IndoBERT	**0.999684**	**0.9997**	**0.99968**	**0.9986854**	**0.9996849**	**0.999686**
CNN-Based	0.9935392	0.9935367	0.993537	0.99377606	0.9935392	0.9935643
Wordformer	*0.997479*	*0.9974791*	*0.997479*	*0.9974944*	*0.997479*	*0.997483*

Best results are marked with **bold,** second best marked with *italic*

A. Benchmark Results

Table I presents the benchmark results of five models evaluated on the spam classification task using Indonesian YouTube comments. Among all models, IndoBERT consistently achieved the best overall performance across every metric, with an accuracy of 0.9997, macro F1 of 0.9997, and near-perfect scores in precision (0.9987), recall (0.9997), and AUC (0.9997). This highlights IndoBERT's capability to handle the informal and noisy nature of real-world spam text in Bahasa Indonesia.

The proposed Wordformer model performed remarkably well, achieving an accuracy of 0.9975, macro F1 of 0.9975, and high scores across all other metrics including precision (0.9975), recall (0.9975), and AUC (0.9975). It ranks second across the board and serves as a strong alternative to heavier transformer models.

The CNN-based model also demonstrated solid results, with an accuracy of 0.9935 and macro F1 of 0.9935, reflecting its effectiveness in learning local spam patterns through convolutional filters. Its precision and recall were slightly lower than those of Wordformer, but it still maintained robust classification quality.

Meanwhile, the traditional machine learning models, Support Vector Machine (SVM) and Random Forest (RF), showed weaker performance. SVM achieved an accuracy of 0.9578 and macro F1 of 0.9577, while RF reached 0.9729 in both metrics. Although these models are computationally lighter, their performance across all metrics was noticeably below that of the deep learning approaches.

Analysis of the confusion matrices shows that IndoBERT commits only a single false positive and a single false negative; Wordformer produces two false positives and three false negatives; the CNN baseline yields four false positives and four false negatives; and even the Random Forest and SVM models keep false positives and false negatives to only six and four samples, respectively—demonstrating that all methods restrict misclassification counts to the low single digits.

B. Computational Complexity

Table II summarizes the computational complexity of each model in terms of parameter count, number of multiply-add operations, and disk size. As expected, IndoBERT is by far the largest model, containing over 66 million parameters, requiring approximately 530 million multiply-adds, and occupying nearly 695 MB of storage. In contrast, the proposed Wordformer model is significantly more efficient, with roughly 9.4 million parameters, 75 million multiply-adds, and a model size of 96.34 MB. The CNN-based model falls between these two in size and compute, with over 3 million parameters and 144 million multiply-adds. Traditional models such as SVM and Random Forest have minimal complexity, with just 4,511 and 11,100 parameters respectively, and model sizes well under 0.1 MB. Multiply-add operations are not reported for these two, as such operations are not typically meaningful or comparable for non-neural models. These results provide a foundational comparison of the resource demands associated with each model architecture.

TABLE II
COMPLEXITY BENCHMARK

Method	# Params	# Multiply Adds (M)	Model Size (MB)
SVM	4511	n/a	0.03
Random Forest	11100	n/a	0.08
IndoBERT	66,660,101	530.53	695.29
CNN-Based	3,173,102	144.56	15.92
Wordformer	9,429,509	75.17	96.34

C. Ablation Study

Table III shows the ablation benchmark between different hyperparameters.

The number of transformer layers significantly impacted performance. The model with only one layer showed the weakest results (Accuracy: 0.9551), while the two-layer version slightly improved across all metrics. The best performance was achieved with three layers, reaching 0.9962 accuracy and 0.9962 macro F1, suggesting that a deeper

TABLE III
ABLATION STUDY

Ablation Study	Accuracy	Weighted F1	Macro F1	Precision	Recall	AUC
Transformer Block Number						
• 1 Layer	0.95511	0.9550	0.95511	0.958	0.95511	0.95512
• 2 Layer	0.9503	0.9503	0.95033	0.9526	0.9503	0.9505
• 3 Layer	0.996	0.996	0.996	0.996	0.996	0.996
Embedded Dimension						
• 100	0.996	0.996	0.996	0.996	0.996	0.996
• 128	0.998	0.998	0.998	0.998	0.998	0.998
Feed Forward Dimension						
• 256	0.985	0.985	0.985	0.985	0.985	0.985
• 512	0.996	0.996	0.996	0.996	0.996	0.996
Number of Heads						
• 2	0.9952	0.9952	0.9952	0.9952	0.9952	0.9952
• 4	0.9960	0.9960	0.9960	0.9960	0.9960	0.9960
Dropout						
• 0.1	0.998	0.998	0.998	0.998	0.998	0.998
• 0.2	0.996	0.996	0.996	0.996	0.996	0.996
• 0.3	0.992	0.992	0.992	0.992	0.992	0.992
Learning Rate						
• 1e-5	0.996	0.996	0.996	0.996	0.996	0.996
• 5e-5	0.998	0.998	0.998	0.998	0.998	0.998
• 1e-4	0.998	0.998	0.998	0.998	0.998	0.998
Optimizer						
• Adam	0.992	0.992	0.992	0.992	0.992	0.992
• Adam Weight	0.996	0.996	0.996	0.996	0.996	0.996
• SGD	0.953	0.953	0.953	0.953	0.953	0.953

encoder enhances contextual representation for spam classification.

In terms of embedding dimension, increasing the embedding size from 100 to 128 led to consistent performance gains across all metrics. The 128-dimensional variant achieved 0.998 accuracy and 0.998 macro F1, highlighting that richer token embeddings help the model better capture nuanced differences in text.

The model with a 512-dimensional feedforward layer outperformed the smaller 256 variant. The best configuration achieved 0.996 accuracy, indicating that a larger internal representation within the transformer improves its capacity to model complex patterns in spam language.

Using 4 attention heads led to slightly better overall performance (Accuracy: 0.9962) compared to 2 heads (0.9952). This suggests that increasing attention granularity can help the model focus more effectively on diverse semantic cues in the input.

A dropout rate of 0.2 produced the most balanced results, outperforming both lower (0.1) and higher (0.3) dropout settings. While too little dropout led to potential overfitting, excessive dropout slightly degraded performance, especially in recall and AUC.

The default learning rate of 1e-4 delivered the best outcome with 0.998 accuracy and 0.998 macro F1, outperforming lower rates like 1e-5 and 5e-5. This suggests that Wordformer benefits from a slightly more aggressive learning schedule when combined with proper regularization.

Among the tested optimizers, Adam with class weighting achieved the best results (0.996 accuracy, 0.996 macro F1), slightly outperforming standard Adam. SGD, on the other hand, lagged behind significantly with reduced precision and overall score, indicating it's less suitable for this model and task.

In summary, the ablation study reveals that Wordformer's performance is sensitive to both architectural and training hyperparameters. Deeper transformer stacks, larger embedding and feedforward dimensions, and moderate dropout contribute positively to the model's effectiveness. Moreover, proper optimizer selection and learning rate tuning play a crucial role in stability and convergence. These findings provide practical insights for future refinement and deployment of lightweight transformer models in spam detection tasks, particularly for low-resource languages like Bahasa Indonesia.

IV. DISCUSSION

The experimental findings underscore the strength of transformer-based models in classifying online gambling spam comments in Bahasa Indonesia. IndoBERT achieved near-perfect performance across all evaluation metrics, affirming its capability to model complex and informal linguistic patterns. Its success is likely due to its extensive pretraining on a large-scale Indonesian corpus, allowing it to grasp nuanced spam signals embedded in noisy user-generated text. However, this high accuracy comes at a significant computational cost, with IndoBERT requiring

over 66 million parameters, 530 million multiply-adds, and occupying more than 695 MB. Such resource demands may be impractical for real-time deployment in low-power or edge environments, highlighting the need for more efficient alternatives.

The proposed Wordformer model successfully bridges this gap by offering a competitive performance profile with far fewer parameters and reduced computational cost. Although it trails IndoBERT slightly in terms of raw accuracy and F1 score, it dramatically reduces the model size to 96 MB and multiply-adds to just 75 million—roughly 7× smaller and 85% less compute-intensive. Wordformer consistently outperforms both the CNN-based model and traditional ML baselines, demonstrating that lightweight transformers can deliver strong contextual understanding without the overhead of full-scale pretrained models. This makes it a compelling candidate for deployment in production environments where resource efficiency is critical, such as on-device moderation tools or serverless processing pipelines.

Further analysis suggests that different model types exhibit varying strengths depending on the characteristics of the input comment. For instance, IndoBERT demonstrates superior performance in handling mixed-language comments and non-standard expressions, likely due to its rich contextual embeddings pretrained on diverse Indonesian corpora. Meanwhile, Wordformer performs reliably across both short and long inputs, but shows slightly better consistency in moderately lengthy comments that contain enough context for self-attention to operate effectively. CNN, on the other hand, often excels in short comments with strong local patterns such as repetitive tokens, emojis, or promotional keywords—scenarios where convolutional filters can capture discriminative n-gram features. Traditional ML models like SVM and RF tend to struggle with symbol-heavy or code-mixed input, highlighting their limitations in generalizing to informal or obfuscated spam content.

The ablation study further illustrates the importance of thoughtful architecture and training configuration. Adding more transformer layers, increasing embedding and feedforward dimensions, and selecting an appropriate dropout rate (specifically 0.2) all contribute to better performance. The learning rate of 1e-4 proved optimal, while alternatives like 1e-5 or 5e-5 led to underfitting. Notably, the choice of optimizer had a marked impact—Adam with class weighting outperformed both plain Adam and SGD, reinforcing the importance of handling class imbalance explicitly in spam classification tasks. These insights emphasize that even within a lightweight architecture, fine-tuning each component can yield substantial improvements.

Interestingly, traditional ML models such as SVM and Random Forest lagged behind, both in accuracy and robustness across metrics like recall and AUC. While they remain attractive for their minimal resource usage (under 0.1 MB in size), their limited ability to capture context or adapt to noisy, informal language restricts their utility in complex tasks like spam detection. Taken together, these findings suggest a clear trend: while large-scale pretrained models like IndoBERT offer unmatched accuracy, carefully designed lightweight transformers like Wordformer strike a more practical balance, delivering strong performance while remaining deployable across a wider range of systems and constraints.

It is important to note that IndoBERT benefits from large-scale pretraining on Indonesian corpora, providing it with a natural advantage over models that must learn word representations from scratch. This contextual understanding explains the performance gap observed between IndoBERT and other models like CNN and Wordformer.

IV. CONCLUSION

This study presented a comparative evaluation of multiple machine learning and deep learning models for the task of detecting Judi Online spam comments in Bahasa Indonesia. Through extensive benchmarking, it was found that transformer-based models significantly outperform traditional ML approaches in both accuracy and generalization. Among them, IndoBERT achieved the highest performance but required substantial computational resources. To address this, we proposed Wordformer, a lightweight transformer architecture specifically designed for efficiency. Wordformer demonstrated competitive performance while maintaining a considerably smaller model size and lower compute cost, positioning it as a practical alternative for deployment in constrained environments.

The ablation study provided deeper insights into the architectural and training components that influence model performance. It highlighted that even compact models benefit from careful tuning of parameters such as the number of layers, embedding size, attention heads, and learning rate. Additionally, the choice of optimizer and regularization strategy significantly affects the learning dynamics, particularly in imbalanced classification settings like spam detection.

For future research, several directions could further enhance this work. First, expanding the dataset to include comments from multiple live streams or platforms could improve model generalization across domains. Second, integrating context-aware features, such as user behavior or temporal patterns during live streams, may provide richer signals for spam detection. Third, exploring knowledge distillation techniques could help transfer IndoBERT's strengths into smaller models like Wordformer more effectively. Lastly, further refining augmentation techniques for Bahasa Indonesia—potentially leveraging language-specific resources or multilingual embeddings—could enhance robustness against adversarial or obfuscated spam content.

One limitation of this study is the inherent difference in input representation between models that rely on pretrained contextual embeddings (such as IndoBERT) and those that do not (such as CNN, Wordformer, SVM, and Random Forest). This gap reflects a real-world trade-off in model selection,

where practitioners must balance between leveraging pretrained resources and managing computational efficiency.

In conclusion, this research demonstrates that effective spam detection in Bahasa Indonesia is feasible with both high-performance and lightweight models. The Wordformer architecture, in particular, offers a promising path forward for developing real-world solutions that balance accuracy, interpretability, and efficiency.

REFERENCES

[1] A. Fahrudin *et al.*, "Online gambling addiction: Problems and solutions for policymakers and stakeholders in Indonesia," *J. Infrastruct. Pol. Dev.*, vol. 8, no. 11, p. 9077, Oct. 2024.

[2] T. N. Dellia Putri Octavia, "Negative Impacts Of Online Gambling Reviewed From The Social Economic And Psychological Perspective In Accordance With Undergoing No. 1 Of 2024 On Second Amendment To Undergoing Number11 Of 2008 On Information And Transactions," in *International Conference Restructuring and Transforming Law*, 2024.

[3] S. Sriyana, "Judi Online: Dampak Sosial, Ekonomi, Dan Psikologis Di Era Digital," *J SOCIOPOLITICO*, Feb. 2025.

[4] A. Kosasih and T. Setiady, "Akibat hukum Artis promosikan situs slot Judi online dampak terhadap masyarakat Dan upaya penanggulangnya," *YUSTISI*, vol. 12, no. 1, pp. 67–78, Feb. 2025.

[5] A. Nurdiansyah and A. S. Kanda, "Bahaya Judi Online : Dampak Sosial, Ekonomi, Dan Kesehatan," *sscj-amik*, vol. 2, no. 1, pp. 305–310, Jan. 2024.

[6] L. Rafiqah and H. Rasyid, "The Dampak Judi Online terhadap Kehidupan Sosial Ekonomi Masyarakat," *Al-Mutharahah*, vol. 20, no. 2, pp. 282–290, Dec. 2023.

[7] A. A. Hadi, A. Zaky, N. Rizqiananda, and B. Unggaran, "Edukasi Bahaya Judi Online Digital Sebagai Upaya Pencegahan Dampak Sosial Dan Ekonomi Bagi Masyarakat Komplek Graha Indah 2 Pamulang," *Krepa*, vol. 3, no. 12, pp. 61–70, Dec. 2024.

[8] A. R. Chrismanto, A. K. Sari, and Y. Suyanto, "Critical evaluation on spam content detection in social media," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 8, pp. 2642–2667, Apr. 2022.

[9] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning--based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, Apr. 2022.

[10] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[11] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[13] L. P. Hung and S. Alias, "Beyond sentiment analysis: A review of recent trends in text based sentiment analysis and emotion detection," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 27, no. 1, pp. 84–95, Jan. 2023.

[14] L. Gong and R. Ji, "What does a TextCNN learn?," *arXiv [stat.ML]*, 18-Jan-2018.

[15] Z. Hou *et al.*, "C-BDCLSTM: A false emotion recognition model in micro blogs combined Char-CNN with bidirectional dilated convolutional LSTM," *Appl. Soft Comput.*, vol. 130, no. 109659, p. 109659, Nov. 2022.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv [cs.CL]*, 02-Oct-2019.

[17] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," *arXiv [cs.CL]*, 26-Sep-2019.

[18] X. Jiao *et al.*, "TinyBERT: Distilling BERT for natural language understanding," *arXiv [cs.CL]*, 23-Sep-2019.

[19] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," *arXiv [cs.CL]*, 06-Apr-2020.

[20] Yaemico, "Deteksi Judi Online." 07-Oct-2024.

[21] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv [cs.CL]*, 25-Aug-2014.

[22] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020.