Enhanced framework for detecting Vietnamese hate and offensive spans

Dinh-Hong Vu[1], **Tuong Le**[2]
[1]Natural Language Processing and Knowledge Discovery Research Group, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[2]Faculty of Information Technology, HUTECH University, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Jun 13, 2025
Revised Jan 1, 2026
Accepted Jan 22, 2026

Keywords:

Hate speech detection
Hated speech dictionary
Natural language processing
Offensive language
Social media
Vietnamese text

ABSTRACT

The rise of hate and offensive content on social media platforms, such as Facebook and Twitter, has emerged as an escalating concern, especially in Vietnam. Consequently, detecting hate and offensive spans in Vietnamese text is an essential area of research. This study introduces ViHateOff, an advanced framework that combines a hated speech dictionary (HSD) automatically constructed from the Vietnamese hate and offensive spans (ViHOS) dataset with the pre-trained language models for Vietnamese (PhoBERT)-large language model to enhance the detection of offensive expressions. The framework functions through two primary modules. First, it constructs an HSD from the ViHOS dataset, which serves as a reference for identifying hate and offensive language in Vietnamese text. Second, the framework integrates the PhoBERT-large language model with HSD, enhancing the detection of harmful words in the input text. Experimental results demonstrate that the proposed framework significantly outperforms existing state-of-the-art (SOTA), achieving an F1-score of 0.8693 on the all-spans subset and 0.8709 on the multiple-spans subset representing relative improvements of over 10% compared to the strongest baseline.

Corresponding Author:

Tuong Le
Faculty of Information Technology, HUTECH University
475A Dien Bien Phu, Ward 25, Binh Thanh District, Ho Chi Minh City, Vietnam
Email: lc.tuong@hutech.edu.vn

1. INTRODUCTION

The rise of internet access and social media has transformed communication, enabled real-time information exchange, and fostered global connectivity. However, it has also facilitated the spread of harmful content such as hate speech and offensive language. Hate speech, as defined by the United Nations, targets individuals or groups based on identity factors like religion, ethnicity, or gender, while offensive language encompasses vulgar, abusive, or derogatory expressions that, although not always classified as hate speech, contribute significantly to a toxic and hostile online environment. Such content not only undermines social cohesion and mutual respect but also negatively affects users’ mental health, increasing stress, anxiety, and depression. Additionally, it can damage the reputation and user base of online platforms, prompting governments worldwide to enforce stricter regulations on content moderation. In the Vietnamese context, detecting hate and offensive language poses additional challenges due to linguistic nuances such as tonal variation, regional dialects, limited annotated data, and frequent use of irony, slang, or sarcasm. Addressing these challenges requires not only advanced machine learning techniques but also a deep understanding of Vietnamese cultural and linguistic subtleties to ensure accurate and context-aware detection.

Deep learning has brought significant advances in the field of natural language processing (NLP), improving computers' ability to understand and use natural language. Deep learning models, particularly recurrent neural networks (RNNs), convolutional neural networks (CNNs), and recently, transformer architectures, have demonstrated exceptional effectiveness in various NLP tasks, such as machine translation [1]–[3], text classification [4]–[8], sentiment analysis [9]–[13], and question answering [14]–[17]. Notably, the emergence of large language models like bidirectional encoder representations from transformers (BERT) [18], generative pre-trained transformers (GPT) [19], pre-trained language models for Vietnamese (PhoBERT) [20] and advanced versions such as GPT-4 has significantly improved the ability to capture context and relationships between words in sentences, enabling deeper understanding of semantics and grammar. Moreover, deep learning enables systems to learn from massive datasets, achieving high accuracy in real-world applications such as chatbots, information retrieval, and hate spans detection. The continuous advancement of deep learning is unlocking new opportunities, positioning NLP as a critical area in artificial intelligence. Recent efforts in hate speech and offensive language detection have focused on reducing misclassification between the two, as demonstrated by Yuan *et al.* [21] with an adversarial debiasing approach that improves multilingual performance. The OffensEval shared tasks (SemEval-2019 and 2020) further advanced the field with offensive language identification dataset (OLID)-based benchmark datasets, where Zampieri *et al.* [22] showed that large language models, while strong, still lag behind top systems. Region-specific work, such as Tamil hate speech detection using term frequency-inverse document frequency (TF-IDF) and transformer models [23], and multiclass, multilabel deep learning for Indonesian tweets [24], reflects growing specialization.

Hoang *et al.* [25] introduced Vietnamese hate and offensive spans (ViHOS), the first human-annotated dataset for identifying hate and offensive spans in Vietnamese comments. Unlike earlier datasets that focus only on flagged keywords, ViHOS captures the underlying ideas and opinions expressed within comments. It includes 26,476 spans from 11,056 comments, with clear splits for training, development, and testing. The dataset also provides detailed annotation guidelines and strategies for handling complex language phenomena such as teencode, metaphors, and puns. ViHOS serves as a valuable resource for improving hate speech detection, especially at the span level in Vietnamese. However, span-level detection accuracy in [25] is still limited, indicating a need for further improvement. Therefore, this study introduces ViHateOff to enhance hate and offensive spans detection in Vietnamese, compared to existing state-of-the-art (SOTA) methods in [25]. The main contributions of this study are as follows: i) data preprocessing in the ViHOS dataset to create hated speech dictionary (HSD), which can be used as a reference dictionary for other hate and offensive spans detection tasks, ii) combining PhoBERT-Large, large language model (LLM) with the HSD helps the model better detect harmful words in input text, and iii) experimental results show that ViHateOff can be applied to detect negative comments on social media or similar platforms.

The remainder of this paper is organized as follows: section 2 presents the proposed ViHateOff framework, including the construction of HSD, feature engineering, and model integration. Section 3 reports experimental results and compares our approach with baseline methods. Section 4 concludes the study and outlines future research directions.

2. METHOD

2.1. Hated speech dictionary

Utilizing the ViHOS dataset, we developed the HSD, which contains a compilation of syllables and words designated as "hated" within this dataset. We assume that syllables/words labeled “hated” in the input sentence are highly likely to indicate hateful content in the model's output. Therefore, this dictionary is utilized to mark “hated” syllables/words in the input sentence, thereby increasing their weight during the model's processing. The dictionary construction process is as follows: first, we transform the span-level labels in the ViHOS dataset into syllable-or word-level labels. If a syllable or word falls within a hateful span, its label is set to 1; otherwise, it is assigned a label of 0. Secondly, we collect all syllables and words labeled as 1 to construct HSD.

Our resulting dictionary contains 2,899 items in syllable form and 3,892 items in word form. This dictionary is derived exclusively from the training dataset to ensure its usage does not influence the model's test results, thereby maintaining objectivity and accuracy in evaluating model performance. Table 1 contains entries illustrating an example of syllables and words included in HSD, focusing on vulgar language, negative emotions, and offensive words.

2.2. The overall architecture

Our proposed framework contains three modules, relation extractor, hated masker and classification modules. The relations between three modules are shown in Figure 1. The ViHateOff framework processes input text using two distinct tokenization methods: word tokenization and syllable tokenization. ViHateOff uses syllable tokenization, which splits the text at whitespace and is denoted by ViHateOffsyllable.

Meanwhile, ViHateOff tokenizes text into Vietnamese words is denoted by ViHateOffword. For instance, the sentence "*phải chăng đây là lý do ông này không muốn cách ly*" (translated as "Maybe this is the reason he doesn't want to be quarantined") would be tokenized into words as "*phải_chăng đây là lý_do ông này không muốn cách_ly*" and then further split into tokens based on whitespace. In summary, the ViHateOffsyllable and ViHateOffword frameworks differ only in their input tokens (syllables versus words), while their underlying model structures remain identical.

Table 1. Example of syllables and words in HSD

STT	Word (in Vietnamese)	Word (in English)	Note	Type
1	*vl*	Slang abbreviation for "vãi l*n"	Vulgar language	Syllable
2	*vét*	Lick up/scrape (derogatory)	Vulgar language	Syllable
3	*chán*	Bored	Negative emotion	Syllable
4	*sủa*	Bark (insulting, like a dog)	Offensive word	Syllable
5	*nát*	Ruined (torn, completely bad)	Offensive word	Syllable
6	*đcm*	f**k	Vulgar language	Syllable
7	*hút*	Inject/use drugs (e.g., heroin)	Offensive word	Syllable
8	*ích_kỷ*	Selfish	Negative emotion	Word
9	*độc_mồm*	Spiteful, sarcastic	Offensive word	Word
10	*thần_kinh*	Nervous, anxiety-related	Offensive word	Word
11	*tổ_cha*	Offensive family-related insult	Offensive word	Word
12	*đô_hộ*	Colonize, oppress	Negative emotion	Word
13	*khinh_bỉ*	Disdain, contempt	Offensive word	Word
14	*dơ_bẩn*	Dirty, filthy	Offensive word	Word

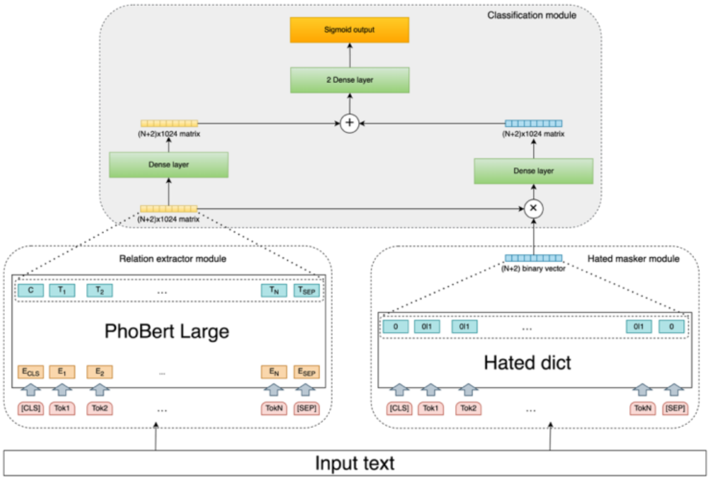


Figure 1. ViHateOff architecture: detecting Vietnamese hate and offensive spans

2.2.1. Relation extractor module

This module utilizes the PhoBERT-large syllable configured to achieve the optimal results reported in [20]. The output of this module is a vector representation for each token in the input text, capturing relational information among tokens. To enhance the model's learning capability, we also incorporate the [CLS] token output to classify the input text as "hated" or not. The label for [CLS] is generated during training as follows: if any token in the text is labeled as "hated", then the label for [CLS] is set to 1; otherwise, it is set to 0. The output of this module is the matrix y_1 with dimensions (N+2)×1024, where N is the maximum number of tokens in the input text. Consistent with [20], we use N=64, along with the [CLS] and [SEP] tokens.

$$y_1 = PhoBert_{large}(X) \tag{1}$$

Where X is the token ID vector generated with the PhoBERT-Large tokenizer, y_1 is output of PhoBERT-Large model.

2.2.2. Hated masked module

This module uses HSD to identify positions of tokens that may be "hated tokens" in the input text. The goal is to double the weight of these tokens in model calculations. The module's output is a binary vector y_2 with (N+2) dimensions, where each element has a value of 1 if its input token in the hated dict else 0.

$$y_2 = Masked(X, D) \quad (2)$$

In this context, *X* is the token ID vector generated with the PhoBERT-large tokenizer, *D* represents the HSD, *y₂* is output vector of masked module. The integration of HSD enhances the model's attention mechanism by assigning greater weights to tokens known to be indicative of hate or offensive intent. Instead of treating all tokens equally, the model leverages prior knowledge from annotated training data to emphasize potentially harmful spans. This mechanism improves detection accuracy in complex or lengthy sentences, and helps the model generalize better to rare or distorted forms of offensive language (e.g., slang or abbreviated expressions), many of which are captured in the dictionary.

2.2.3. Classification module

This module combines the outputs of the two previous modules using two separate dense layers (DL_1 and DL_2). DL_1 handles classification based on the output of the relation extractor module, while DL_2 focuses on classification using tokens in HSD. This is achieved by multiplying the output matrix from the relation extractor with the binary vector from the hated masked module, retaining only vectors for tokens found in HSD. The outputs from these two dense layers provide two distinct classification perspectives on the same data. We then merge these perspectives by adding the two matrices, resulting in an (N+2)×1024 matrix to create the following y_3.

$$y_3 = DL_1(y_1) + DL_2(y_1 \odot y_2) \quad (3)$$

Where y_1, y_2 are result of (1) and (2), *DL* stands for dense layer, ⊙ is element-wise multiplication with broadcasting: each row *i* in y_1 is scaled by $y_2[i]$. Then the result serves as input to the two *DL* and output layer with a sigmoid activation function (σ) to classify whether a token is hated or not.

$$\hat{y} = \sigma(DL_4(DL_3(y_3))) \quad (4)$$

Where y_3 is result of (3), *DL* stands for dense layer. The input token is classified as a hated token if $\hat{y}_i > 0.5$, and as a non-hated token if $\hat{y}_i \leq 0.5$. Although we have not conducted an ablation study to isolate the dictionary's impact, the superior performance of ViHateOff compared to baselines using the same backbone (PhoBERT) implies that the dictionary plays a significant role. Future work will include detailed statistical validation (e.g., p-values and confidence intervals) and ablation experiments to quantify this contribution more precisely.

3. RESULTS AND DISCUSSION

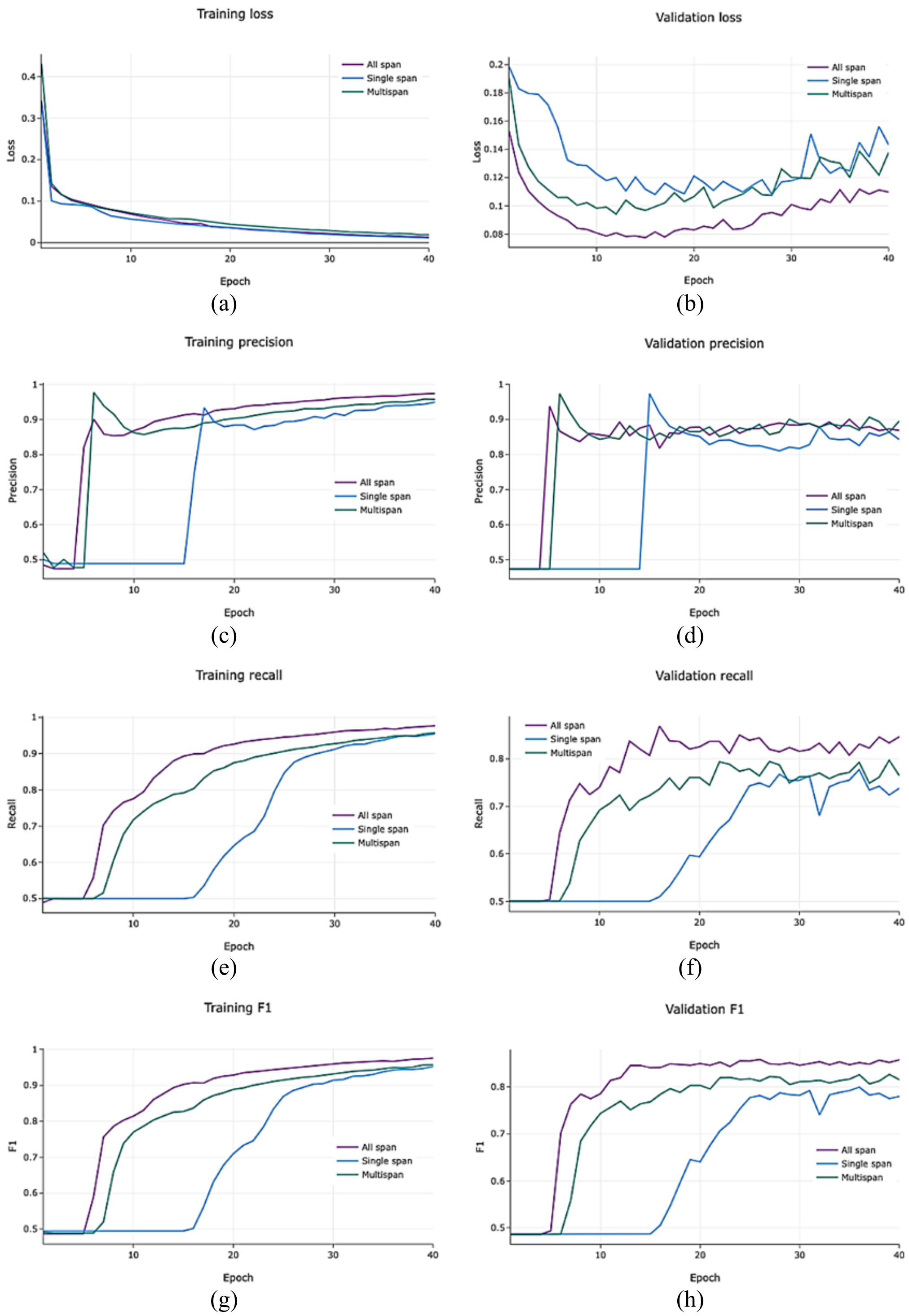
3.1. Experimental settings

This study trained the ViHateOff model using the Adam optimizer with a learning rate of 6×10^{-5}, a batch size of 128, and 40 training epochs. The optimal epoch thresholds were selected for the three subsets: single span, multiple spans, and all spans (details in section 3.2). ViHateOff utilizes the pre-trained PhoBERT-Large model [20] via the hugging face library, with input data from the ViHOS dataset pre-tokenized accordingly. Training was performed on a high-performance system (126 GB RAM, 80 CPUs). For comparison, we adopted the baseline models and results from [25]. Both syllable-based and word-based tokenization approaches were tested (ViHateOff$_{\text{syllable}}$ and ViHateOff$_{\text{word}}$), and evaluated using macro F1-score, precision, and recall, consistent with [25].

3.2. Experiments and results

3.2.1. Experiment 1-determining the optimal number of training epochs for ViHateOff$_{\text{syllable}}$

This experiment, conducted specifically on ViHateOff$_{\text{syllable}}$ as shown in Figure 2, aims to identify the ideal number of epochs for training the model by observing trends over a 40-epoch run. Figure 2(a) and 2(b) shows the training and validation loss over 40 epochs. While training loss steadily declines, it plateaus after epoch 25. Validation loss, however, increases after epoch 18 and sharply rises post-epoch 30, indicating overfitting. Thus, training beyond 30 epochs offers little benefit and may harm generalization. In Figures 2(c) and 2(d), validation precision plateaus around epoch 26, stabilizing near 0.84. The all-span subset exhibits the most stable and highest performance across all metrics. Figures 2(e) to 2(h) (recall and F1-score) follow similar trends, confirming the model's strength on more complex multi-span data. Conversely, for the single span subset, performance remains low in the first 15 epochs, with F1 around 0.33, and only improves slightly by epoch 26. Validation loss increases after epoch 18, indicating overfitting risk. Therefore, the optimal training range for this subset is between epochs 18-25. Epoch 22 was selected for final evaluation, as it provides the best performance across the "single span," "multiple spans," and "all spans" subsets.



(a)

(b)

(c)

(d)

(e)

(f)

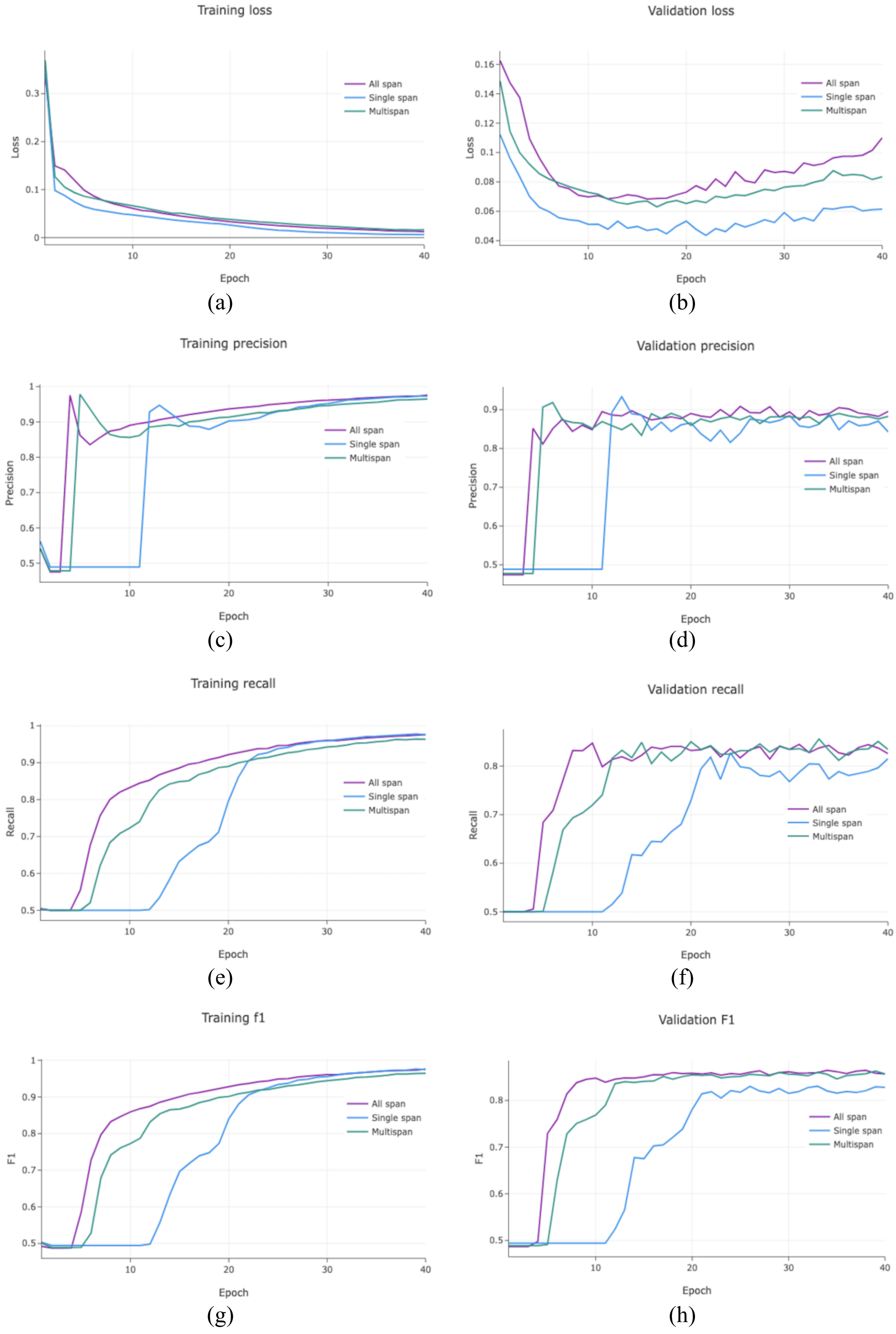
(g)

(h)

Figure 2. Training/validation metrics for ViHateOffsyllable of (a) training loss, (b) validation loss, (c) training precision, (d) validation precision, (e) training recall, (f) validation recall, (g) training F1-score, and (h) validation F1-score

3.2.2. Experiment 2-determining the optimal number of training epochs for ViHateOff$_{\text{word}}$

This study focuses on ViHateOff$_{\text{word}}$, as shown in Figure 3, to determine the best number of epochs for training. By analyzing trends over the course of 40 epochs, we assess the model's learning efficiency and identify potential signs of overfitting. In this experiment, similar trends to experiment 1 were observed, with the validation loss starting to rise after epoch 18 (Figures 3(a) and 3(b)) while other metrics, such as precision (Figures 3(c) and 3(d)), recall (Figures 3(e) and 3(f)), and F1-score (Figures 3(g) and 3(h)) stabilized around epoch 26. This indicates that the model could start to overfit if training proceeds beyond this point. Therefore, the optimal training range for this subset appears to be between 18 and 25 epochs, as it strikes a balance between effective learning and preventing overfitting. For consistency with the previous experiment, epoch 22 was selected, providing the best performance across the "single span," "multiple spans," and "all spans" subsets.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3. Training/validation metrics for ViHateOffword of (a) training loss, (b) validation loss, (c) training precision, (d) validation precision, (e) training recall, (f) validation recall, (g) training F1-score, and (h) validation F1-score

3.2.3. Experiment 3-comparison with state-of-the-art models

In this experiment, we evaluate our framework, trained using the optimized parameters outlined earlier, against the SOTA models for detecting hate and offensive spans. Table 2 shows that the BiLSTM-CRF+Pho2W model performed poorly, with the lowest F1-scores: 0.3594 (word) and 0.4329 (syllable), highlighting its limitations on the "single span" subset. In contrast, modern models like PhoBERT and cross-lingual language model-RoBERTa (XLM-R) showed better performance, especially the proposed ViHateOff$_{\text{word}}$, which achieved the highest F1-score (0.8174) and recall (0.8427), with an average rank of 1.33. ViHateOff$_{\text{syllable}}$ also led in precision (0.8519) within the syllable-based group and ranked second overall. These results confirm the superiority of the ViHateOff framework in both syllable and word-level span detection.

Table 3 highlights the superior performance of ViHateOff$_{\text{word}}$ on the "multiple spans" subset, achieving the best scores in precision (0.8597), recall (0.8830), and F1-score (0.8709), with an average rank of 1.00. ViHateOff$_{\text{syllable}}$ also performed well, ranking second across all metrics with an average rank of 2.00. In contrast, BiLSTM-CRF+Pho2W showed the weakest performance in both input types, confirming its limitations in detecting multiple spans. While PhoBERT and XLM-R models performed competitively, they remained behind ViHateOff. Notably, PhoBERT$_{\text{large}}$ outperformed PhoBERT$_{\text{base}}$, and XLM-R$_{\text{large}}$ slightly surpassed XLM-R$_{\text{base}}$ in all key metrics.

Table 4 summarizes model performance on the "all spans" subset. ViHateOff$_{\text{word}}$ achieved the best results across all metrics—precision (0.8823), recall (0.8573), F1-score (0.8693)—with the top average rank of 1.33. ViHateOff$_{\text{syllable}}$ also performed strongly, ranking second overall with the highest recall (0.8611) and an F1-score of 0.8678. In contrast, BiLSTM-CRF+Pho2W models showed the weakest performance, especially the word-based version with the lowest F1-score (0.7036) and average rank of 8.00. Modern models like PhoBERT$_{\text{large}}$ and XLM-R$_{\text{large}}$ outperformed their base counterparts but remained behind ViHateOff, confirming the advantage of task-specific and Vietnamese-optimized architectures.

Table 2. Performance comparison of experimental methods on "single span" subsets

	Model	P (rank)	R (rank)	F1 (rank)	Average rank
Syllable	BiLSTM-CRF+Pho2W$_{\text{syllable}}$	0.4222 (7)	0.5009 (7)	0.4329 (7)	7.00
	XLM-R$_{\text{base}}$	0.7604 (3)	0.7653 (3)	0.7203 (4)	3.33
	XLM-R$_{\text{large}}$	0.7577 (4)	0.7679 (2)	0.7214 (3)	3.00
	ViHateOff$_{\text{syllable}}$	**0.8519 (1)**	0.7492 (5)	0.7907 (2)	2.67
Word	BiLSTM-CRF + Pho2W$_{\text{word}}$	0.3196 (8)	0.4468 (8)	0.3594 (8)	8.00
	PhoBERT$_{\text{base}}$	0.7392 (6)	0.7485 (6)	0.7016 (6)	6.00
	PhoBERT$_{\text{large}}$	0.7435 (5)	0.7567 (4)	0.7067 (5)	4.67
	ViHateOff$_{\text{word}}$	0.7957 (2)	**0.8427 (1)**	**0.8174(1)**	**1.33**

Table 3. Performance comparison of experimental methods on "multiple spans" subsets

	Model	P (rank)	R (rank)	F1 (rank)	Average rank
Syllable	BiLSTM-CRF+Pho2W$_{\text{syllable}}$	0.5134 (7)	0.5712 (7)	0.5068 (7)	7.00
	XLM-R$_{\text{base}}$	0.7203 (6)	0.7927 (3)	0.7327 (4)	4.33
	XLM-R$_{\text{large}}$	0.7829 (4)	0.7569 (4)	0.7357 (3)	3.67
	ViHateOff$_{\text{syllable}}$	0.8433 (2)	0.8473 (2)	0.8453 (2)	2.00
Word	BiLSTM-CRF+Pho2W$_{\text{word}}$	0.3533 (8)	0.5001 (8)	0.4013 (8)	8.00
	PhoBERT$_{\text{base}}$	0.7761 (5)	0.7329 (6)	0.7092 (6)	5.67
	PhoBERT$_{\text{large}}$	0.7878 (3)	0.7557 (5)	0.7321 (5)	4.33
	ViHateOff$_{\text{word}}$	**0.8597** (1)	**0.8830** (1)	**0.8709** (1)	**1.00**

Table 4. Performance comparison of experimental methods on "all spans" subsets

	Model	P (rank)	R (rank)	F1 (rank)	Average rank
Syllable	BiLSTM-CRF +Pho2W$_{\text{syllable}}$	0.7452 (7)	0.7769 (5)	0.7453 (7)	6.33
	XLM-R$_{\text{base}}$	0.7766 (6)	0.7574 (7)	0.7467 (6)	6.33
	XLM-R$_{\text{large}}$	0.8071 (3)	0.7887 (3)	0.7770 (3)	3.00
	ViHateOff$_{\text{syllable}}$	0.8748 (2)	**0.8611 (1)**	0.8678 (2)	1.67
Word	BiLSTM-CRF + Pho2W$_{\text{word}}$	0.6823 (8)	0.7489 (8)	0.7036 (8)	8.00
	PhoBERT$_{\text{base}}$	0.7870 (5)	0.7680 (6)	0.7569 (5)	5.33
	PhoBERT$_{\text{large}}$	0.8028 (4)	0.7835 (4)	0.7716 (4)	4.00
	ViHateOff$_{\text{word}}$	**0.8823 (1)**	0.8573 (2)	**0.8693 (1)**	**1.33**

The results across all three tables consistently confirm the superiority of the ViHateOff frameworks, especially ViHateOff$_{\text{word}}$, which achieved the highest metrics and lowest average ranks across all span types.

ViHateOff$_{\text{syllable}}$ also performed strongly, consistently ranking second and highlighting the value of syllable-level processing. These findings demonstrate that leveraging advanced, Vietnamese-specific deep learning techniques as done in ViHateOff is crucial for achieving SOTA performance in detecting hate and offensive spans.

Beyond evaluation metrics, we also assessed the inference efficiency of the ViHateOff framework under different deployment scenarios. On a CPU-only machine (80 vCPUs, 126 GB RAM), the model achieves approximately 10-15 comments per second, suitable for batch processing or offline moderation. On a high-performance server with an NVIDIA A100 GPU, throughput reaches up to 900-1,200 comments per second in batch mode, or 100-200 comments per second for real-time settings. These results indicate that ViHateOff is feasible for large-scale deployment, particularly when GPU resources are available.

3.3. Error analysis

During our evaluation, we identified common sources of misclassification in the ViHateOff framework, primarily due to linguistic complexities inherent in Vietnamese social media language. The key error categories include: emerging or unlisted slang expressions not captured in the current HSD (e.g., "*ăn hành*", "*gãy kèo*", "*nổ*" depending on context); sarcastic or ironic expressions, which require deep contextual understanding and often reverse the apparent sentiment. Another category is metaphorical or indirect insults, where offensive meaning is implied rather than directly stated (e.g., seemingly positive phrases used in mocking tone like "*đỉnh thật*").

For instance, in the sentence "*ôi bố cái lũ thanh niên hãm lol. đẹp mặt quá*"/"oh father of those disgusting youths [vulgar slang] how glorious…", the gold labels include "*bố, cái, lũ, hãm, lol, đẹp, mặt, quá*" but the model only identified "*lũ, hãm, lol*" as hateful tokens. Tokens "*bố, cái*" carry a contextual insulting tone, but are not inherently offensive on their own. Tokens "*đẹp, mặt, quá*" is sarcastic, reversing the literal sentiment (literally "*so glorious*") into ridicule challenging for the model to detect without pragmatic understanding.

These errors highlight the limitations of both the static dictionary approach and current model architecture in handling nuanced or context-dependent language. In future work, we propose to: continuously update the HSD with new slang and social media terms. Incorporate sarcasm-aware pretraining or fine-tune on datasets specifically annotated for irony. Integrate context-sensitive sentiment models to improve robustness against indirect expressions.

4. CONCLUSION

This study proposed an enhanced framework for detecting hate and offensive spans in Vietnamese by addressing key linguistic challenges through a combination of data preprocessing, the construction of a HSD, and the application of the PhoBERT-Large language model. The experimental results on the ViHOS dataset demonstrate that our approach consistently outperforms SOTA methods across all evaluation subsets: single span, multiple spans, and all spans. These findings confirm the robustness and effectiveness of the framework in identifying harmful content, particularly on social media platforms. Despite the demonstrated effectiveness of the HSD, its coverage remains inherently limited due to its static nature and reliance on manually annotated data. As slang and offensive expressions evolve rapidly on social media, many newly emerging terms are not captured in the current dictionary. Furthermore, the model struggles with sarcasm, irony, and metaphorical language, where the literal meaning often differs from the intended sentiment. To address these challenges, we plan to develop an automated dictionary update pipeline, integrate sarcasm-aware pretraining and context-sensitive sentiment models, and collect additional annotated datasets to enhance the adaptability and robustness of the system in real-world scenarios. In future work, we plan to enrich the dataset by collecting more labeled data from diverse sources to improve the model's generalizability. Additionally, incorporating advanced techniques such as hyperparameter tuning, ensemble strategies, and external knowledge bases may further enhance the adaptability and performance of the system in real-world applications.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Dinh-Hong Vu	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Tuong Le		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study did not involve human subjects, personal information, or identifiable data requiring informed consent.

ETHICAL APPROVAL

This study did not involve human subjects, animals, or any experiments requiring institutional review board approval, as it is based solely on computational analysis using publicly available datasets and pre-trained language models.

DATA AVAILABILITY

The dataset used in this study is publicly available in GitHub at https://github.com/phusroyal/ViHOS.

REFERENCES

[1] S. Ranathunga, E. S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023, doi: 10.1145/3567592.

[2] S. Zhu, C. Mi, T. Li, Y. Yang, and C. Xu, "Unsupervised parallel sentences of machine translation for Asian language pairs," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 3, pp. 64:1-64:14, 2023, doi: 10.1145/3486677.

[3] P. N. Nguyen and P. Tran, "Constructing a Chinese-Vietnamese bilingual corpus from subtitle websites," *International Journal of Intelligent Information and Database Systems*, vol. 16, no. 4, pp. 385–408, 2024, doi: 10.1504/IJIIDS.2024.141748.

[4] D. H. Vu, K. Nguyen, K. T. Tran, B. Vo, and T. Le, "Improving fake job description detection using deep learning-based NLP techniques," *Journal of Information and Telecommunication*, vol. 9, no. 1, pp. 113–125, 2025, doi: 10.1080/24751839.2024.2387380.

[5] Y. Wu and J. Wan, "A survey of text classification based on pre-trained language model," *Neurocomputing*, vol. 616, 2025, doi: 10.1016/j.neucom.2024.128921.

[6] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and K. Akinwolere, "Text classification: how machine learning is revolutionizing text categorization," *Information*, vol. 16, no. 2, 2025, doi: 10.3390/info16020130.

[7] N. A. Saputra, K. Aeni, and N. M. Saraswati, "Indonesian hate speech text classification using improved k-nearest neighbor with TF-IDF-ICSpF," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 21–30, 2024, doi: 10.15294/sji.v11i1.48085.

[8] G. Z. Nabiilah, I. N. Alam, E. S. Purwanto, and M. F. Hidayat, "Indonesian multilabel classification using IndoBERT embedding and MBERT classification," *International Journal of Electrical & Computer Engineering*, vol. 14, no. 1, 2024, doi: 10.11591/ijece.v14i1.pp1071-1078.

[9] T. K. Tran, H. M. Dinh, and T. T. Phan, "Building an enhanced sentiment classification framework based on natural language processing," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 1771–1777, 2022, doi: 10.3233/JIFS-219278.

[10] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022, doi: 10.1007/s10462-022-10144-1.

[11] H. Murfi, T. Gowandi, G. Ardaneswari, and S. Nurrohmah, "BERT-based combination of convolutional and recurrent neural network for Indonesian sentiment analysis," *Applied Soft Computing*, vol. 151, 2024, doi: 10.1016/j.asoc.2023.111112.

[12] L. Zhu, H. Zhao, Z. Zhu, C. Zhang, and X. Kong, "Multimodal sentiment analysis with unimodal label generation and modality decomposition," *Information Fusion*, vol. 116, 2025, doi: 10.1016/j.inffus.2024.102787.

[13] N. Darraz, I. Karabila, A. El-Ansari, N. Alami, and M. El Mallahi, "Integrated sentiment analysis with BERT for enhanced hybrid recommendation systems," *Expert Systems with Applications*, vol. 261, 2025, doi: 10.1016/j.eswa.2024.125533.

[14] Y. Zhuang, Y. Yu, K. Wang, H. Sun, and C. Zhang, "ToolQA: a dataset for LLM question answering with external tools," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 50117–50143.

[15] P. Lu *et al.*, "Learn to explain: multimodal reasoning via thought chains for science question answering," in *Proceedings of the 36th International Conference on Neural Information Processing System*, 2022, pp. 2507–2521.

[16] K. V Tran, H. P. Phan, K. V Nguyen, and N. L. T. Nguyen, “ViCLEVR: a visual reasoning dataset and hybrid multimodal fusion model for visual question answering in Vietnamese,” *Multimedia Systems*, vol. 30, no. 4, 2024, doi: 10.1007/s00530-024-01394-w.
[17] S. Chowdhury and B. Soni, “R-VQA: a robust visual question answering model,” *Knowledge-Based Systems*, vol. 309, 2025, doi: 10.1016/j.knosys.2024.112827.
[18] 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 American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, vol. 1, pp. 4171–4186.
[19] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018. [Online]. Available: https://openai.com/research/language-unsupervised
[20] D. Q. Nguyen and A. T. Nguyen, “PhoBERT: pre-trained language models for Vietnamese,” in *Proceedings of EMNLP Findings*, 2020, pp. 1037–1042, doi: 10.18653/v1/2020.findings-emnlp.92.
[21] S. Yuan, A. Maronikolakis, and H. Schütze, “Separating hate speech and offensive language classes via adversarial debiasing,” in *Proceedings of Sixth Workshop on Online Abuse and Harms (WOAH)*, 2022, pp. 1–10, doi: 10.18653/v1/2022.woah-1.1.
[22] M. Zampieri, S. Rosenthal, P. Nakov, A. Dmonte, and T. Ranasinghe, “OffensEval 2023: offensive language identification in the age of large language models,” *Natural Language Engineering*, vol. 29, no. 6, pp. 1416–1435, 2023, doi: 10.1017/S1351324923000517.
[23] R. Rajalakshmi, S. Selvaraj, and P. Vasudevan, “Hottest: hate and offensive content identification in Tamil using transformers and enhanced stemming,” *Computer Speech & Language*, vol. 78, 2023, doi: 10.1016/j.csl.2022.101464.
[24] A. Gandhi *et al.*, “Hate speech detection: a comprehensive review of recent works,” *Expert Systems*, vol. 41, no. 8, 2024, doi: 10.1111/exsy.13562.
[25] P. G. Hoang, C. Luu, K. Q. Tran, K. V Nguyen, and N. L.-T. Nguyen, “ViHOS: hate speech spans detection for Vietnamese,” in *Proceedings of EACL*, 2023, pp. 652–669, doi: 10.18653/v1/2023.eacl-main.47.

BIOGRAPHIES OF AUTHORS

Dinh-Hong Vu received the B.S. degree in Information Technology from the VNU Ho Chi Minh City University of Science, Ho Chi Minh City, Vietnam, in 2005, and the M.Sc. degree in Computer Science from the VNU Ho Chi Minh City University of Science, in 2011. He is currently a researcher with NLP-KD Research Group, Ton Duc Thang University, Vietnam. His research interests include natural language processing, machine translation, and text clustering. He can be contacted at email: vudinhhong@tdtu.edu.vn.

Tuong Le received his Ph.D. degree in Computer Science from Sejong University, Korea, in 2020. He is currently a researcher and faculty member at the Faculty of Information Technology, HUTECH University, Vietnam. He has authored over 40 publications in high-impact journals, including Information Sciences, Expert Systems with Applications, IEEE Access, and Engineering Applications of Artificial Intelligence. His research interests span machine learning, imbalanced learning, deep learning, business intelligence, data analysis, data mining, and pattern mining. He can be contacted at email: lc.tuong@hutech.edu.vn.