Neural Computing and Applications (2023) 35:573–594
https://doi.org/10.1007/s00521-022-07745-w

ORIGINAL ARTICLE

Vietnamese hate and offensive detection using PhoBERT-CNN and social media streaming data

Khanh Quoc Tran[1,2] · An Trong Nguyen[1,2] · Phu Gia Hoang[1,2] · Canh Duc Luu[1,2] · Trong-Hop Do[1,2] · Kiet Van Nguyen[1,2]

Received: 18 January 2022 / Accepted: 17 August 2022 / Published online: 17 September 2022

Abstract
Society needs to develop a system to detect hate and offense to build a healthy and safe environment. However, current research in this field still faces four major shortcomings, including deficient pre-processing techniques, indifference to data imbalance issues, modest performance models, and lacking practical applications. This paper focused on developing an intelligent system capable of addressing these shortcomings. Firstly, we proposed an efficient pre-processing technique to clean comments collected from Vietnamese social media. Secondly, a novel hate speech detection (HSD) model, which is the combination of a pre-trained PhoBERT model and a Text-CNN model, was proposed for solving tasks in Vietnamese. Thirdly, EDA techniques are applied to deal with imbalanced data to improve the performance of classification models. Besides, various experiments were conducted as baselines to compare and investigate the proposed model's performance against state-of-the-art methods. The experiment results show that the proposed PhoBERT-CNN model outperforms SOTA methods and achieves an F1-score of 67.46% and 98.45% on two benchmark datasets, ViHSD and HSD-VLSP, respectively. Finally, we also built a streaming HSD application to demonstrate the practicality of our proposed system.

Keywords Hate speech detection · Sentiment analysis · Transformer · Streaming data

1 Introduction

Along with the advances in technology of the Fourth Industrial Revolution, the rapid rise of social networks has been astoundingly altering our daily life. In that situation, safety in cyberspace is an issue that directly affects the user's life, especially objects such as children or vulnerable people. According to Mohan et al. reports [1], the social media environment in which many harmful contents such as hateful comments, fake news, contents that violate community standards influence not only on the large proportion of users but also on online moderators. Hate speech is typically described as any communication that disparages a person or group based on any attribute such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other trait. The following are some examples of hateful and offensive comments posted on Vietnamese social media:[1] cứ phải chửi cho mới ch u im :)))$_{\text{you only shut your mouth until I swear at you:))))}}$; lũ chó đói$_{\text{those hungry dogs}}$; hài vcl$_{\text{so fucking funny}}$

However, censoring hate and offensive comments on social media faces many challenges because of their enormous volume and variety in both magnitude and topics. According to Suha Abu et al., research [2], 293,000 posts

✉ Khanh Quoc Tran
18520908@gm.uit.edu.vn

An Trong Nguyen
18520434@gm.uit.edu.vn

Phu Gia Hoang
19520215@gm.uit.edu.vn

Canh Duc Luu
19521272@gm.uit.edu.vn

Trong-Hop Do
hopdt@uit.edu.vn

Kiet Van Nguyen
kietnv@uit.edu.vn

[1] Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam

[2] Vietnam National University, Ho Chi Minh City, Vietnam

[1] The several examples in this article are given to demonstrate the seriousness of the hate speech problem. They are based on actual online data and do not reflect the authors' opinions.

are posted every 60 s on the billion-user social networking platform, Facebook, and over 510,000 comments are written there. Moreover, according to the prestigious statistics reporting site, Statista [3], Facebook had to remove more than 11.3 million pieces of offensive and hateful content globally in 2018. In 2019, YouTube also removed over 1.800 million comments that violated community standards, and these statistics have risen dramatically on both platforms. In 2020, Facebook must remove more than 81 million hateful and offensive posts, a seven times increase from 2018. While YouTube must remove over 4800 million comments in 2020, this tripled the figures in 2019.

The above results are the efforts of the two most prominent social networking platforms to stop hate and offense in their platforms globally. According to a report from the Wall Street Journal [4], up to 2018, Facebook had to spend hundreds of millions of dollars on their content moderation teams. Besides, the major American technology information site—The Verge, reports that Google also has a team consisting of nearly 10,000 people responsible for the same task. However, this team still has many shortcomings.

Firstly, despite the fact that Facebook is available in over 100 languages [4], but only slightly more than half of them have dedicated content censorship teams. Meanwhile, Southeast Asia (including Vietnam) is a significant market for Facebook but still lacks human resources with advanced language skills. In addition, the social network environment in Vietnam is extremely toxic, according to a survey by Microsoft [5].

Secondly, even though content censorship teams have been trained and informed about extremely hateful content, they need to deal with its consequences. Many of them have psychological disorders, and some even face Post-traumatic Stress Disorder (PTSD) [6], which is prevalent after witnessing a heinous crime, and many are reported that they will never fully recover.

Thirdly, the large corporations that own these social networks and research laboratories have allocated resources to develop systems using Artificial Intelligence to solve this dilemma, but they are still impractical. These systems utilize rich and quality data sources that originated from their social networking platforms. They possess leading-edge methods that can be applied across multilingual [7], making powerful systems that quickly classify hate and offensive content. However, it is difficult for these systems to recognize the content that lacks context, specialized culture native, and are slow to keep up with the continuously improving development in hateful content. Furthermore, these systems have not been studied extensively and deeply enough in Vietnamese due to Limitations remaining in building the systems which result in being unusable to solve this task.

Therefore, our research contributes to HSD task in Vietnamese. Our research proposes a novel system that applies advanced natural language processing techniques to classify the hate and offensive comments on social networks towards a healthier, safer online space. Our research can handle issues as small as single comments to as vast as continuously processing enormous amounts of data in real-time. The main scientific contributions of our research are summarized as follows.

1. We implemented strict and efficient data pre-processing techniques to clean comments collected from social networking sites. These techniques enhance the data quality and significantly improve extracting information before training the model.
2. A novel HSD model is proposed to improve the performance of the task in Vietnamese. To this end, the various experiments were conducted with four state-of-the-art approaches: machine learning, deep learning, transfer learning, and combined learning. Compared to our proposed PhoBERT-CNN model, these approaches aid in developing the baseline models.
3. We applied EDA techniques to the ViHSD dataset and the HSD-VLSP dataset to deal with imbalanced data and verify the effectiveness and necessity of data augmentation for the Vietnamese HSD task.
4. To demonstrate the usefulness of the proposed system, we built an application that continuously streams data

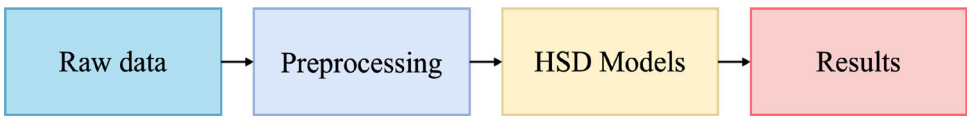


Fig. 1 An overview of existing approach for hate speech detection

Table 1 Number of changes on the datasets after the Phase 1

Datasets	Lowercase	Redundant		Inconsistent		Link
		Spaces	Characters	Unicode	Accented words	
ViHSD	28,540	488	2127	753	620	21
HSD-VLSP	0	1	2667	0	761	1

from the massive data source of social media platforms to detect hate and offensive comments.

Our proposed system can be applied to online newspapers or websites having small comments quantity but require strict censorship, as well as huge social networks or forums. The system helps to improve the comprehensive censorship of offensive and hateful comments on cyberspace in Vietnam. We contribute to building a positive, civilized environment or satisfying the need to orient and protect vulnerable subjects such as the elderly and children. Moreover, the application is also a basis for agencies and organizations to evaluate and control management, psychological research, and education behaviors.

The rest of the paper is organized as follows. In Sect. 2, we survey and describe an overview of the fundamentals of the HSD task and relevant studies. Our proposed approach is presented in detail through Sect. 3. Section 4 is our experimental results on the given datasets, contribution of each module in the PhoBERT-CNN model, and instructions and actual performance of the HSD application with streaming data. Finally, Sect. 5 is Conclusion and Future works.

2 Fundamental of hate speech detection on streaming data

2.1 A brief introduction to hate speech detection task

HSD and sentiment analysis have been inextricably linked [8], and these tasks have recently become popular topics in Natural Language Processing. In this section, we summarize the Vietnamese HSD task [9, 10]. This task aims to detect whether a comment on social media is HATE, OFFENSIVE, or CLEAN. Formally, the task is described as follows.

Input: Given Vietnamese comments on the social networks sites.

Output: One of three different labels is predicted by classifiers.

- *Hate speech (HATE)* contains abusive language, which regularly bears the aim of insulting individuals or groups and might include hate speech, derogatory and offensive language. An item (post or comment) is identified as HATE if it (1) targets individuals or groups based on their characteristics; (2) demonstrates a transparent intention to incite harm or to market hatred; (3) may or might not use offensive or profane words.
- *Offensive but not hate speech (OFFENSIVE)* is an item that could contain offensive words, but it does not target individuals.
- *Neither offensive nor hate speech (CLEAN)* is a normal item. It is conversation, and expresses emotions normally, and it does not contain offensive language or hate speech is a normal item.

2.2 Data pre-processing in HSD

Data pre-processing techniques always play an essential role in data classification tasks from Vietnamese social networks in general and hate speech detection tasks in particular [11]. Khang et al. [12] investigated the impact of pre-processing on datasets collected from Vietnamese social networks. According to the findings of this study, pre-processing has a significant impact on extracting information from data. Vietnamese comments on social media frequently contain characters and words associated with emotional undertones presenting in various ways, making it difficult to identify, differentiate, and extract information. Khang et al. [12] also succeeded in using initial data pre-processing to improve results by 4.66%. This is a success compared to earlier work on the same datasets and evaluation metrics.

However, current studies on both the ViHSD and the HSD-VLSP datasets have not used modern and effective data pre-processing techniques to improve the performance of classification models. Only simple pre-processing techniques, such as word-segmenting texts into words, lower case text, removing or anonymizing sensitive information, and removing URLs and non-alphabetic characters, were utilized in previous studies. In our study, we inherit the advantages of previous studies and implement novel and specific pre-processing techniques to handle some of the most difficult challenges with social network data, such as SOTA Vietnamese word segmentation using VnCoreNLP [13], De-teencode, and stopwords removal (see Sect. 3.2). These techniques enhance the performance of models.

2.3 Existing HSD models

Some surveys on hate speech and machine learning for HSD give the research information on the current state of this field. They not only provide a structured overview of previous approaches [14] but also describe key sub-areas that have been explored to recognize these types of utterances automatically [8]. In addition to providing a survey of modern natural language processing (NLP) techniques used for automatic detection of hate speech on social networks online, Alrehili et al. [15] indicated that preprocessing techniques such as Bag of Words, Dictionary, Part of

Speech, and machine learning models such as Random Forest, Naive Bayes, and Decision Trees also produce positive results for the HSD task. This motivates other researchers, such as Waseem et al. [16], Chen et al. [17], and Davidson et al. [18], to apply the HSD system with the purpose of solving the real-life problem of hate speech on social network.

On the other hand, we conducted a survey on the related works serving the task of classifying comments on social networks in Vietnamese, especially the Vietnamese Hate Speech Detection task is still modest [9, 10, 19–24]. Specifically, the current studies revolve only based on two typical datasets by their outstanding high quality and large quantity of data points: ViHSD [10] and HSD-VLSP [9] datasets.

Methods for solving HSD problems are numerous, with machine learning models being the most fundamental. Support Vector Machine and Random Forest models applied in the study of Davidson et al. [18] and Martins et al. [25] are the best approaches across their studies, and the results serve as the foundation for future development of other methods. In recent years, SOTA solutions with exceptional performance have emerged, such as the development of single models for multi-language such as BERT [26], RoBERTa [27], XLM-R [28], and combinations to create more superior models such as those such as BERT-CNN [29], RoBERT-CNN [30], XLMR-CNN [31], which provide opportunities for HSD performance increase.

As inspired by the success of combining variant BERT with the CNN model [29–31], the PhoBERT-CNN combined model is implemented in this work to investigate its efficacy in the task of Vietnamese HSD.

CNN is used instead of other typical deep neural networks such as LSTM [32], Bi-LSTM [33], and GRU [34] since it is currently the most successful model for addressing short text classification tasks [35]. The convolution and pooling techniques of CNN aid in the extraction of the main concepts and keywords of the text as features, resulting in a significant improvement in the performance of the classification model. However, the CNN network has a significant limitation that it is not suitable for sequence-level text [35, 36]. To address this limitation, the large-scale pre-trained language model for Vietnamese PhoBERT is the appropriate combination due to the fact that the PhoBERT has a duty on extracting features from sentences for the input of the Text-CNN model.

PhoBERT, the first large-scale monolingual pre-trained language model for Vietnamese, was introduced by Nguyen et al. [37]. PhoBERT was trained on about 20 GB of data, including approximately 1 GB from the Vietnamese Wikipedia corpus and the rest of 19 GB from the Vietnamese news corpus. The architecture of PhoBERT is similar to the RoBERTa model developed by Liu et al. at Facebook [27] (the model is optimized from the BERT model with a large amount of data training data up to 160 GB and a 10× increase over BERT). Furthermore, when it comes to Vietnamese, PhoBERT has been demonstrated to perform and produce better results than the current best multilingual model [37], namely the XLM-R model [28].

2.4 Hate speech detection with streaming data

Besides new datasets and methodologies, numerous applications and systems for real-time data processing are also introduced. Some typical projects can be mentioned, such as data streaming process of Nagarajan et al. [38], and real-time tweets processing on Twitter, which uses Spark Streaming [39]. Some real-life applications have been introduced in the last decade, in 2015, Burnap et al. [40] successfully provided Application Programming Interfaces (APIs) based on website services such as CrowdFlower or Amazon Mechanical Turk which can be integrated into a data pipeline in order to classify hate speech text. Then in 2018, Anagnostou et al. also presented a web application for actively reporting YouTube [41].

However, to the best of our knowledge, current research on HSD for Vietnamese is still at the theoretical analysis stage, and no practical application solutions have been introduced. Therefore, we need a highly scalable, reliable, and fault-tolerant data streaming engine for real-time HSD.

We also overcome the remaining restrictions on the ViHSD and HSD-VLSP datasets in previous studies [9, 10, 19–24] by proposing a two-phase data pre-processing techniques. Furthermore, we inherit the advantages of each study, such as the ability to conduct the experiment with deep learning, transfer learning, and combined models. In particular, we combine a robust pre-trained model for Vietnamese, PhoBERT and a deep learning model, Text-CNN. At the end of the experiment, an HSD application with streaming data was deployed to demonstrate the usefulness and contribution of our proposed system.

3 Proposed hate speech detection system for social media streaming data

3.1 Proposed system architecture

This section proposes our efficient and straightforward approach for the Vietnamese HSD task. By fine-tuning strategies, we aim to optimize the combined model, PhoBERT-CNN, for providing the best-performance model. Figure 2 shows the overview of the system using four essential components: the pre-processing techniques, the data augmentation techniques, the core method PhoBERT-CNN, and the HSD streaming application.

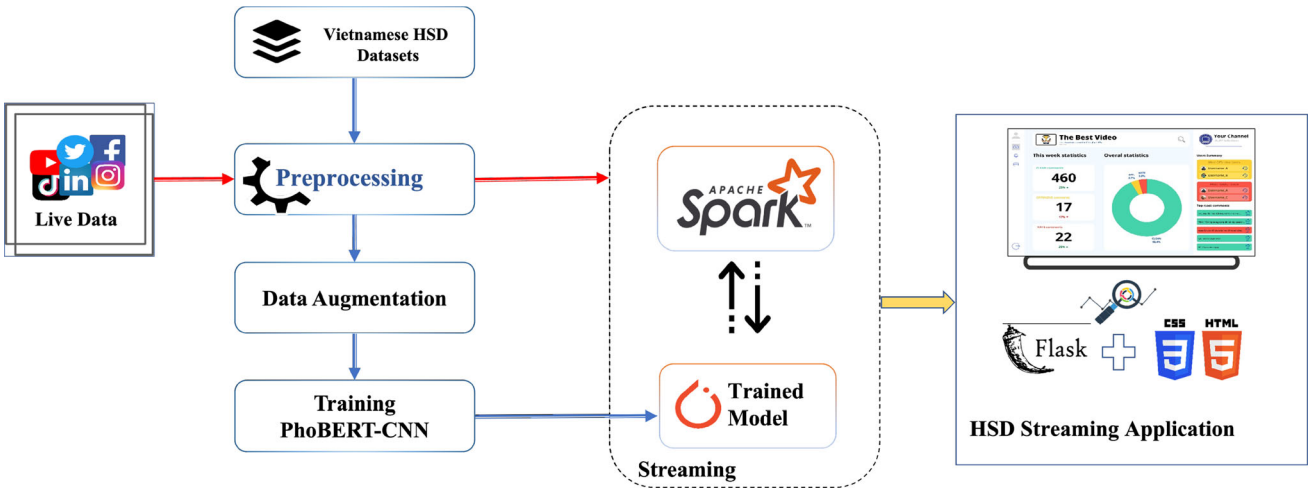


Fig. 2 Our proposed approach for Vietnamese hate speech detection

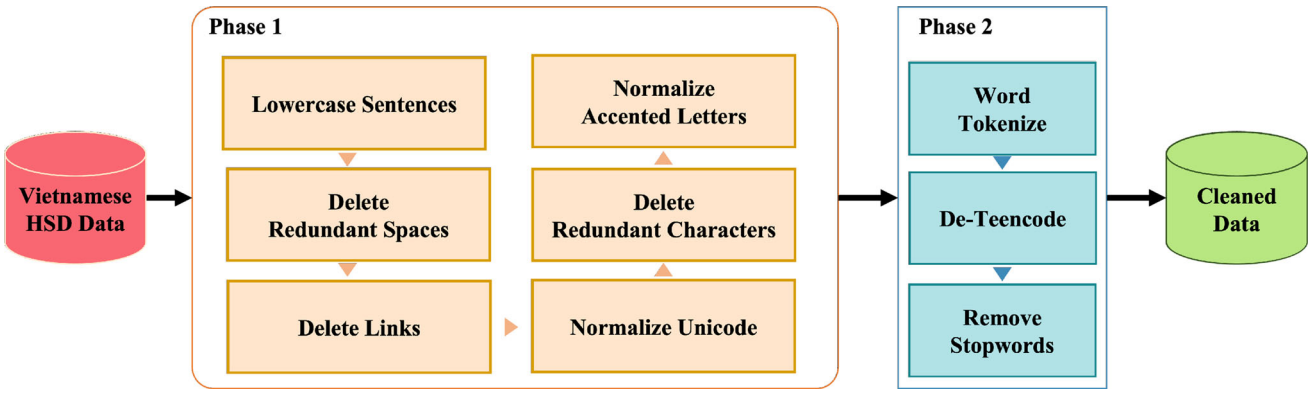


Fig. 3 Data preparation steps

3.2 Data pre-processing

We use two datasets: ViHSD [10] and HSD-VLSP [9] which contain 33,400 and 20,345 comments, respectively. Because the ViHSD and HSD-VLSP datasets are collected from social networking sites, they contain highly complex and diverse comments. Especially, abundant comments in both datasets contain, non-unicode standard characters, teen code, acronyms, and words with repeating characters. Therefore, we proceed to build a data pre-processing process to improve the quality of the datasets to extract valuable features before using them for training the classification models. Figure 3 depicts an overview of the two-phase data pre-processing procedure.

3.2.1 Phase 1

Lowercase sentences: All the characters of all the comments in the datasets are lowercase. We do this to avoid Python seeing two exact words as separate because of their capital letters.

Delete redundant space: Users on social media unwittingly or wittingly type multiple spaces on their comments. Therefore, we have decided to remove those redundant spaces.

Delete links: We believe that website in comments do not affect the sentiment of the comment. As a result, we have also decided to remove them all.

Fig. 4 An example of a comment which generally means "If have nothing to say, have a call to see each other is happy enough. So fucking simple" and deleted redundant characters version of it

Normalize unicode: We also see a lot of Vietnamese words in the dataset that are the same, but Python detects them as separate because of their Unicode. The reason for this is that numerous Unicode Transformation Formats (UTF) are commonly used, so we choose to normalize data to UTF-8.

Delete redundant characters: We remove redundant characters that the users intentionally make.

Figure 4 above is an instance of the process of Deleting Redundant Characters. The word "vuiiii", which means happy, and the word "vllllll", which is an offensive teen code in Vietnamese but does not have a clear meaning in English, both after removing redundant characters, they become "vui" and "vl". However, the duplicated characters in the word "kk", which is also a teen code for the action of laughing - kaka, and "call" are "k" and "l", respectively, are not removed. Indeed, the word "call" will become "cal", which has no meaning, and the word "kk" will become "k", which is a different teen code meaning no in English.

Normalize accented letters: Because of the inconsistency in placing diacritical marks, we decide to normalize them in comments by followings rules:

- If there is one vowel, the diacritical mark(s) will be on that vowel. For example: má (mom), lá (leaf), mê (love).
- If there are two vowels, the diacritical marks will be on the first one. For example: lóa (shinny), qùa (gift). If three vowels or two vowels follow with a consonant, the diacritical marks will be on the second vowel, for example: khuỷu (elbow), quán (store).
- "ê" and "ơ" are exceptional because the diacritical marks are always on them, for example: khuy n (dog), quở (reproachfully).

All the above steps in Phase 1 are conducted in the same order. The output of this Phase 1 is fed directly to the next Phase 2.

3.2.2 Phase 2

Word tokenize: The input sentence is split into words or meaningful word phrases. In order to do this, we used Word Segmenter of VnCoreNLP [13] for the PhoBERT model and NLTK [42] for the other models. Because the comments in both datasets, ViHSD [10] and HSD-VLSP [9], are raw text data, word segmentation is required to prepare the data for PhoBERT model training [37]. Moreover, PhoBERT used the VnCoreNLP RDRSegmenter [13] to pre-process the pre-training data [37] (including Vietnamese word and sentence segmentation), it is recommended that the same word segmenter must be used for PhoBERT downstream applications in relation to the input raw texts. On the other hand, the other models could learn from text data at the token level without requiring word segmentation, as the PhoBERT model does. As a result, we decided to tokenize the pre-training data using NLTK [42].

Table 2 Examples of teencodes and their expansions

No.	Teencode	De-Teencode	
		Vietnamese sentences	English meanings
1	đc đ y	được đ y	nice
2	ko	không	no
3	cc	con c*c	d*ck

De-teencode: In social networks, people usually spend a significant amount of their time to chit chat and also often use the short form of words to type faster. Some are used to trick the systems when they are swearing. Moreover, those abbreviations also have their name in Vietnam, teen codes. As a result, to help our models better understand the input sentences, we had to map those teen codes into their original words. Furthermore, the process of mapping teen codes, we named it De-teencode, and the following Table 2 shows some instances of them.

Remove stopwords: We also removed stopwords from the comments because of their meaninglessness. In our experiments, we used the Vietnamese stopword dictionary [43] to remove stop words in the sentence.

In Phase 2, the data are tokenized, De-teencode, and removed stopwords. Phase is in that order since the output of Word tokenizers is a list of words, word phrases, and characters that are separate from the others by space. Those characters then are checked if they are teen code and will be De-teencode in the next step. Therefore, the De-teencode step following after the Word tokenize step is a wise decision. Finally, after the De-teencode step, we remove all stopwords, and the reason we remove stopwords after De-teencode step is that those teencodes possibly are also stopwords. Table 3 presents basic statistics about teencode and stopword words in two datasets, ViHSD and HSD-VLSP. From the statistics, we can observe that the relatively high proportions of the terms teencode and stopword in the two data sets indicate the use of several acronyms and abbreviations by social media users.

3.3 Dealing with imbalanced data

More and more effective solutions for the HSD task based on pre-trained language models have recently been proposed. The development of various large-scale pre-trained language models is driving up demand for high-quality, large-scale data sources. One of the most important

Table 3 The number of teencodes and stopwords of the datasets

Datasets	Teencodes		Stopwords		#Words
	Frequency	Percentage (%)	Frequency	Percentage (%)	
ViHSD	15,344	4.00	153,330	40.01	383,270
HSD-VLSP	13,757	3.24	127,531	30.01	424,301

priorities in this context is the distribution of data samples, particularly the data balance, which has a significant impact on model evaluation performance [44]. However, there is a significant difference in the number of CLEAN comments compared to OFFENSIVE and HATE comments in many real-world situations, particularly in the two experimental datasets, ViHSD and HSD-VLSP, resulting in a skewed sample distribution. Learning algorithms will be biased toward the majority group due to the above problem, making this a challenging and exciting task to deal with in this paper. Meanwhile, the minority classes are typically more beneficial in terms of information mining, as they contain crucial information for the task of HSD despite its quite rare. To address this challenge, we focused on developing an intelligent system that used efficient data preprocessing techniques, suitable data augmentation, and a robust classification model to overcome bias. This is known as learning from imbalanced data [45].

As a result, inspired by Wei et al. [46], we intend to apply EDA techniques with four operations: synonym substitution, random insertion, random swap, and random deletion to deal with imbalanced data. These techniques are applied to the ViHSD training set and the HSD-VLSP dataset in this paper with the percentage of replacement word in the sentence (α) equal to 0.15.

We first applied the EDA techniques on the ViHSD training set and the entire original HSD-VLSP dataset to enhance the data on HATE and OFFENSIVE labels. Table 4 describes the information about the HSD-VLSP dataset after making data augmentation.

Table 4 shows that after applying EDA techniques, the number of data and vocabulary size on the HATE and OFFENSIVE labels grew dramatically. Figure 5 depicts the label distribution in the ViHSD training set and the HSD-VLSP dataset before and after augmentation. When compared to the original dataset, the data on the augmented dataset are more distributed evenly.

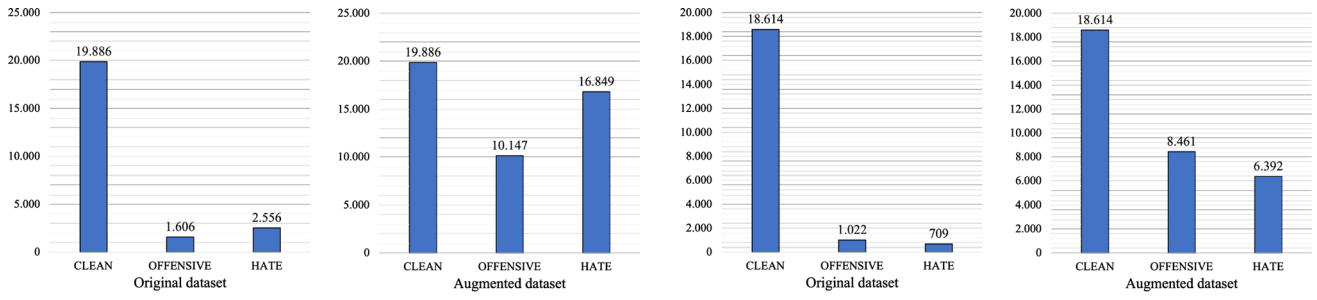
3.4 Proposed PhoBERT-CNN model for HSD

Variant BERT and CNN combined models have recently been widely used to classify short text collected from social networks, particularly to classify hate and offensive comments that achieve promising results [29–31]. As a result, in this work, the PhoBERT and CNN combined model is deployed to evaluate their efficacy in classifying hate and offensive comments for Vietnamese. The combined PhoBERT-CNN model is expected to significantly improve the classification performance thanks to the resonance mechanism of the two single models, reducing errors between the predicted labels and the actual labels. Both of the two single models, PhoBERT and Text-CNN, outperformed other models on the tasks of classifying Vietnamese text, particularly on the ViHSD and HSD-VLSP datasets [9, 10, 19–24]. Figure 6 presents the architecture of our approach for Vietnamese HSD.

Firstly, we present the architecture of PhoBERT [37] and how the PhoBERT model is adapted to perform as a word embedding layer to extract information from data. The PhoBERT model was chosen because it outperforms previous monolingual and multilingual pre-trained language model approaches, achieving new state-of-the-art performance on four downstream Vietnamese NLP tasks, including Vietnamese Hate Speech Detection [9, 10, 19–

Table 4 The ViHSD training set and HSD-VLSP dataset overview statistics before and after data augmentation

Dataset	Label	Original dataset			Augmented dataset		
		Num. comments	Avg. word length	Vocab. size	Num. comments	Avg. word length	Vocab. size
ViHSD training set	CLEAN	19,886	6.55	130,238	19,886	6.55	130,238
	OFFENSIVE	1606	7.24	11,624	10,147	7.57	76,802
	HATE	2556	12.08	30,883	16,849	11.64	196,086
HSD-VLSP	CLEAN	18,614	14.85	276,557	18,614	14.85	276,557
	OFFENSIVE	1022	8.87	9063	8461	8.05	68,093
	HATE	709	14.23	10,087	6392	13.41	85,713



(a) ViHSD training set. **(b)** HSD-VLSP dataset.

Fig. 5 The labels distribution in the ViHSD training set and the HSD-VLSP dataset before and after augmentation

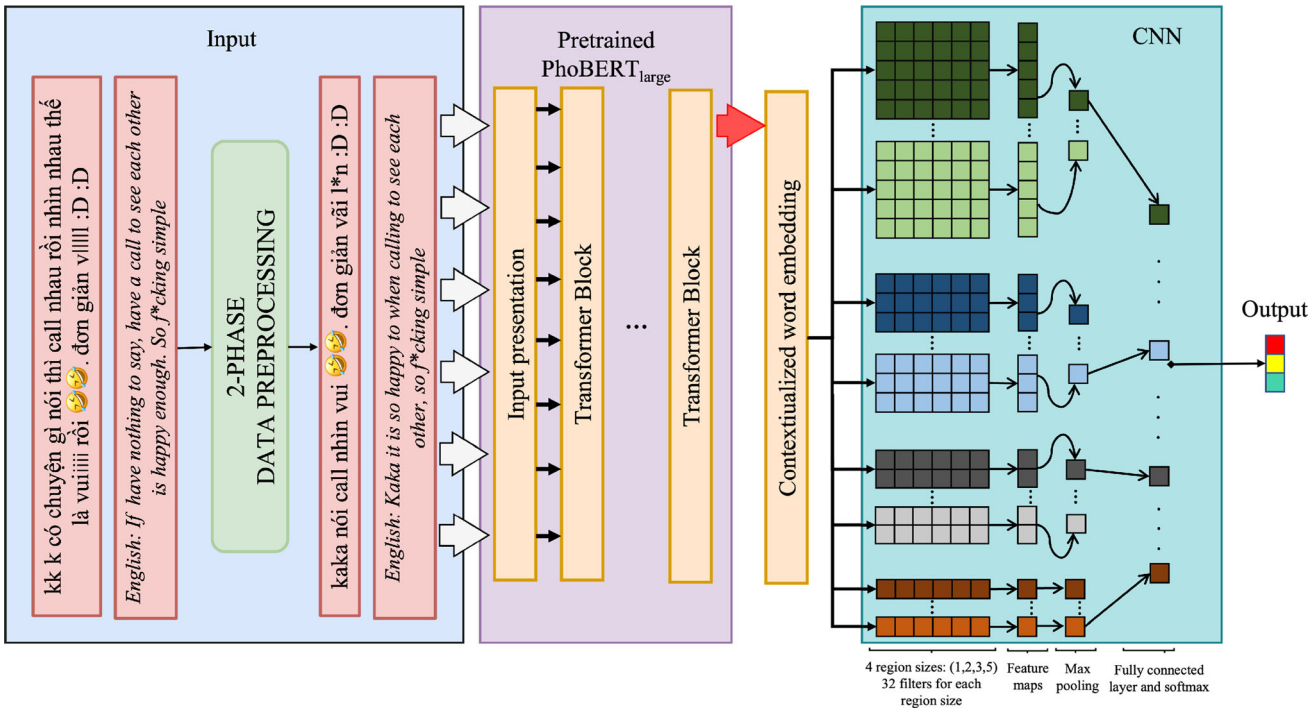


Fig. 6 An overview of our Vietnamese HSD system using PhoBERT-CNN

24]. The architecture of the PhoBERT model is a multilayer architecture comprised of multiple layers of Bidirectional Transformer encoder. It takes the representation of a text sentence composed of a string by the contextualized words as input. The input representation of the PhoBERT model is built by summing those tokens with the segment vectors and the corresponding positions of the words in the sequence. Figure 7 depicts an example of the input data representation of PhoBERT model.

- We use Positional Embeddings with a maximum sentence length of 20 words.
- The first word for each string defaults to the special word `[CLS]`. The output of the final hidden state corresponding word `[CLS]` will be used to represent the whole sentence in the classification.
- When a string only comprises a single sentence, the embedding segment can be applied directly to that sentence.
- If the string contains more than two sentences, sentences are distinguished in two steps: sentences are separated by a special token called `[SEP]` and independent segment embedding for each sentence is added.

Next, the Fully-connected layer at the end of the pre-trained PhoBERT model is replaced with a CNN network architecture [36]. Because CNN is currently the most successful model for solving short text classification tasks [35], it is utilized instead of other typical deep neural networks such as LSTM, Bi-LSTM, and GRU [47–49].

In this research, we construct the layers of the CNN model as follows:

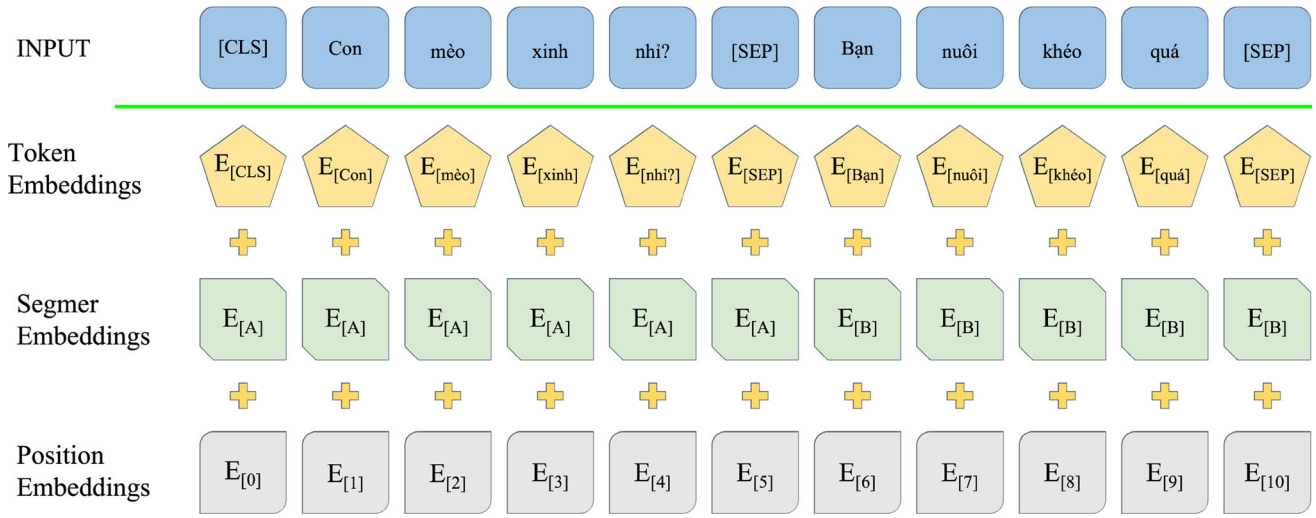


Fig. 7 The process of representing the input of PhoBERT model

- *INPUT*: The input layer is used to initialize the input object from a matrix of vectors. The input and output objects of this layer have dimensions equal to the number of dimensions of the vector matrices.
- *CONV1D*: We build four convolution layers using the Conv1D layer to extract features from a matrix of vectors. For each Conv1D, we all use a filter of a particular size, set kernel size values, and use the ReLU activation function to enhance convergence.
- *POOLING*: We perform maximal pooling (Max Pooling) using the MaxPool1D function. The output of this layer is a matrix of features that have been reduced in size but still retain the features extracted from the previous Conv1D layer.
- *DROPOUT*: Before constructing the dropout layer, we use the torch.cat() function to concatenate the feature matrices that have been Pooled from the previous layer. Then, we set dropout values to remove random notes from each hidden layer.
- *FC*: In this layer, we construct a loss function and an optimized function to connect input data from the dropout layer to the classification classes. We utilize the Adam optimization algorithm and the CrossEntropy loss function (Eq. 1) for optimization:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{1}$$

where M is the number of classes (CLEAN, OFFENSIVE, HATE), log is the natural log, y is binary indicator (0 or 1) if class label c is the correct classification for observation o, p is predicted probability observation o is of class c.

The convolution and pooling techniques of CNN aids in the extraction of the main concepts and keywords of the text as features, resulting in a significant improvement in the performance of the classification model. However, the CNN network has a significant limitation which is not suitable for sequence-level text [35, 36]. To address this limitation, the large-scale monolingual language model pre-trained for Vietnamese PhoBERT is the appropriate combination due to the fact that the PhoBERT has a duty on extracting features from sentences for the input of the Text-CNN model. Following that, a contextualized word embedding of comments from PhoBERT is fed into the Text-CNN model to get the feature maps. Finally, the prediction labels are given through a softmax layer.

The combined PhoBERT-CNN model will significantly improve the classification performance thanks to the resonance mechanism of the two single models, reducing errors between the predicted labels and the actual labels (see Sect. 4.6.5).

3.5 Integrating HSD model to streaming processing system

One of the top-priority requirements is an advanced method for handling hate and offensive comments in big data environments such as social networks. This helps social networks in general and social networks in Vietnam, identify hate and offensive comments better, and also reduce the workload of moderators. Furthermore, the media agencies need an automatic moderator tool to more precisely monitor the comments that are permitted to be displayed.

We built an application for comment analysis capable of continuously collecting content from social networking sites to analyze comment nuances to meet these needs. These

platforms will provide an API to send processing comments requests to the classification application.

3.5.1 Streaming data ingestion

Data streaming is the process of collecting data continuously in real-time from multiple data sources that are often put into stream processing applications to get essential insights.

Data streaming is critical for dealing with enormous amounts of live data. Such data can come from various sources, especially social networking sites like Facebook, Youtube, and Twitter.

Several real-time data streaming approaches are available, including Apache Kafka, Spark Streaming, and Apache Flume. In this paper, we will implement data streaming using Spark Streaming [50].

This section presents an end-to-end architecture on how to stream data from social media platforms, clean them, and apply the combined PhoBERT-CNN model to detect the hate or offense of each data. Figure 8 depicts an overview of the Vietnamese hate speech detection with a streaming system based on the PhoBERT-CNN model.

Input data: Live data collected through streaming API.

Main model: Data pre-processing and hate speech detection on the collected data.

Output: A parquet file with all the data and their hate speech prediction.

3.5.2 Instructions streaming processing for HSD

We successfully constructed a system capable of handling large amounts of data in real-time from social networking platforms, especially from YouTube comments, by conducting surveys and experiments on processing streaming data [38–41]. This section presents the instructions of the system in practice.

Part 1: Send comments from the Youtube Data API:

In this part, we authenticate and connect to the Youtube Data API using our developer credentials. First, we need to set up the essential information to log in and utilize the Youtube Data API, such as DEVELOPER_KEY, YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION. Next, in the QUERY section, we change the URL to query the video that needs to be processed. In addition, parameters such as textFormat to set the format of return comments, maxResults to set the number of comments are limited to each session. We also build a TCP socket between the Youtube Data API and Spark [50], which waits for the Spark Streaming call and delivers data.

Part 2: Data pre-processing and hate speech detection:

Comment data collected and stored through Youtube Data API will be transmitted to the primary system to perform pre-processing according to the process described in Sect. 3.2.

After pre-processing, the normalized and high-quality data are used to predict their labels using the combined PhoBERT-CNN model. We use SparkSQL to query and visualize how Spark Streaming [50] organizes and presents predictions relative to comments. The prediction results in 0.0, 1.0, and 2.0 are corresponding to the CLEAN, OFFENSIVE, and HATE labels , which indicate the polarity of nuance provided by the comments.

Finally, the query results are converted to DataFrame format so that the administrator can observe and monitor it more efficiently and are stored in a parquet file. These findings will be used to assist administrators in deciding whether to delete comments containing hate or offensive content or not.

3.5.3 Reproducibility of the Proposed HSD System

We also provide a solution constructed in PyTorch for reproducibility to make it easy to apply our novel proposed system for Vietnamese HSD. The code for our trained models and streaming application is publicly available for research purposes at https://github.com/khanhtran0412/ViSoMeCens.

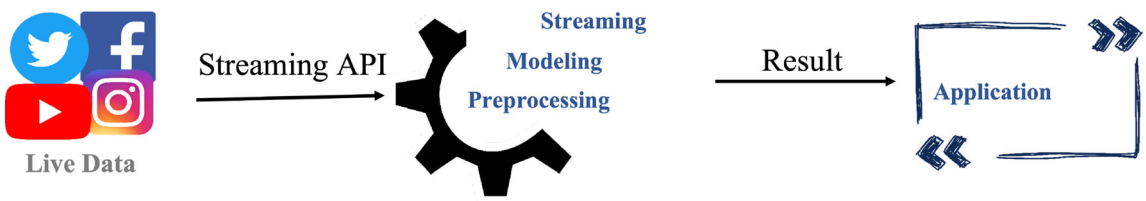


Fig. 8 An overview of our Vietnamese HSD streaming system

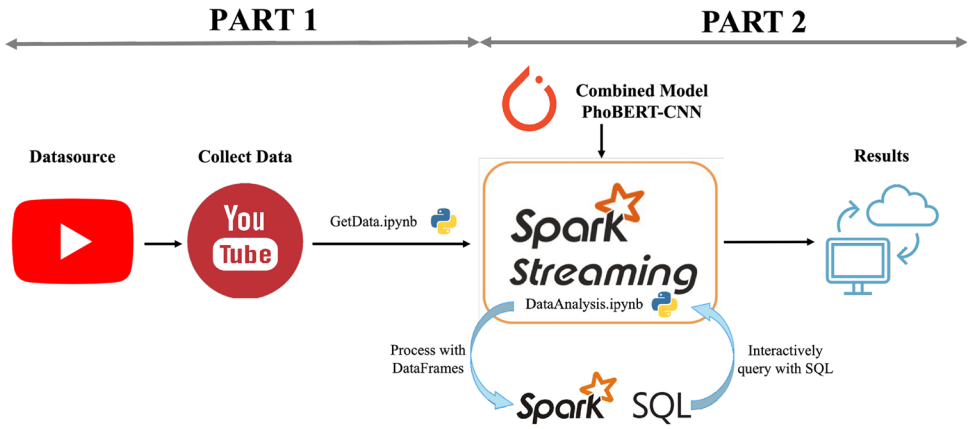


Fig. 9 The end-to-end architecture of our system using spark streaming

4 Experiments and results

4.1 Experimental procedure

This section provides an outline of how we conducted experiments in order to propose a new and efficient HSD system for Vietnamese. Firstly, the hate speech data is collected from social media, especially the two datasets ViHSD and HSD-VLSP will be cleaned using the data preprocessing process as described in Sect. 3.2. Accordingly, the data after cleaning and quality assurance is used to train our baselines and proposed model. With each HSD model conducted, we fine-tune the hyperparameters to find the optimal hyperparameters and help improve the performance of the model. We also apply data augmentation techniques to the ViHSD training set and HSD-VLSP dataset to address the challenging problem of data imbalance. Next, we evaluated the performance of the models we conducted using the F1-macro and Accuracy metrics. The model evaluation results are described in detail in Sect. 4.6. Based on the achieved results in the F1-score, we choose the model with the best performance to conduct an error analysis on the wrong predictions discovered in our system.

In addition, comparisons with previous studies were made to assess the development of our study. Besides, an ablation analysis was carried out to investigate the effectiveness and contribution of our proposed PhoBERT-CNN approach. At the end of the experiment, a pilot experiment was performed to evaluate the actual performance of the proposed system when dealing with hate speech detection from data streaming in which our system was deployed to a social network, especially Youtube. Figure 10 presents an overview of the experimental procedure in this paper.

4.2 Baseline models for HSD performance comparison

4.2.1 Machine learning approach

Multinomial Naive Bayes: This algorithm predicts and classifies data based on observable data and statistics, using the Bayes theorem of probability theory [51–53].

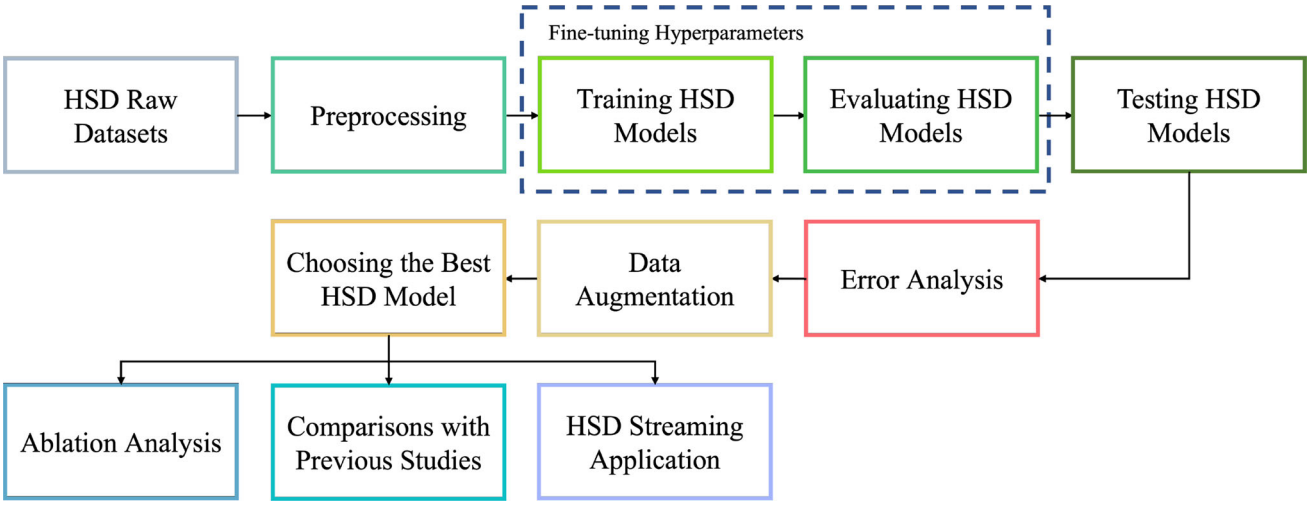


Fig. 10 Overview of the experimental procedure for Vietnamese HSD

Multinomial Naive Bayes is a supervised learning algorithm that is commonly used in machine learning because it is relatively easy to train and achieve high performance.

Logistic regression: Logistic regression is a binary classification algorithm, it is a simple, well-known, and important method in the discipline of machine learning. In addition, this algorithm is also used in a machine learning application to classify incoming data based on previous data.

By analyzing the relationship between all of the existing independent variables, the Logistic Regression model predicts a dependent data variable. In natural language processing, this method requires manual features extracted from data for text classification [18, 22, 54, 55].

Decision tree: Decision tree is a supervised learning algorithm, it is the most powerful and popular method for classification [56–58]. Decision tree algorithm is also known as a structure tree, where each node represents a test on an attribute, each branch is an outcome of the test, and each leaf node is a class label.

This approach used basic rules from training data to predict the class or value of the target variable. Specifically, the record started from the root of the tree and compared the attribute with the node attribute at each branch in the decision tree before predicting a final class label in the leaf node.

Random forest: Random Forest is a Supervised learning method used to solve classification and regression tasks. It is built on multiple sets of Decision Tree and the output of this algorithm is based on the aggregate decision on the decision trees it generates with the voting method.

However, we cannot understand how this algorithm works due to the complicated structure of this model and this is one of the Black Box methods [18, 59–61].

4.2.2 Deep learning approach

Convolutional neural network (text-CNN): A convolutional neural network (CNN) is a multistage Neural network architecture developed for classification [36]. By using convolutional layers, it can detect combination features. Our experiments employ four convolutional layers with 32 filters for each layer. Finally, the softmax function uses the result to predict the label for the text.

Bidirectional long short-term memory (Bi-LSTM): The Bi-LSTM [33] is a famous variant of RNN [62]. The Bidirectional Long Short Term Memory can be trained using all available input information within the past and way forward for a selected timeframe. This method is robust in classification problems, and most of its achieved high-performance classification results. Therefore, we plan to choose it to compare with other classification models during this task.

4.2.3 Transfer learning approach

The transfer learning model has attracted increasing attention from NLP researchers worldwide for its outstanding performance. One of the SOTA language models as BERT, which stands for Bidirectional Encoder representations from transformers, is published by Devlin et al. [26]. BERT and its variations (BERTology) [63–65] such as RoBERTa [27], XLM-R [28], PhoBERT [37] have almost dominated and asserted their strength on natural language processing tasks, even for Vietnamese hate speech detection tasks [10, 21, 22]. For the aforementioned reasons, we decided to use BERT and its variants to find the optimal solution with good performance and contribute to the successful construction of our proposed solution.

BERT [26]: is a contextualized word representation model pre-trained using bidirectional transformers and based on a masked language model. In this work, we use the train set to fine-tune the pre-trained BERT model before classifying comments or posts from websites or social networks.

Robustly optimized BERT approach (RoBERTa) [27]: is trained with dynamic masking, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples. RoBERTa, implemented in PyTorch, modifies key hyperparameters in BERT, including removing BERT's next-sentence pretraining objective and training with much larger mini-batches and learning rates.

XLM-RoBERTa (XLM-R) [28]: is a multilingual model trained using over two terabytes of cleaned and filtered CommonCrawl data. Upsampling low-resource languages during training and vocabulary generation, generating a more extensive shared vocabulary, and raising the overall model capacity to 550 million parameters are all important contributions of XLM-R.

PhoBERT [37]: For Vietnamese, the SOTA method was first released and called PhoBERT by Nguyen et al. [37] for solving Vietnamese NLP problems. PhoBERT is a pre-trained model, which has the same idea as RoBERTa, a replication study of BERT is released by Liu et al. [27], and there are modifications to suit Vietnamese.

4.2.4 Combined approach

BERT and CNN combined models have recently been widely used to classify short text collected from social networks, particularly to classify hate and offensive comments and achieves promising results [29–31]. In this paper, variant BERT and CNN combined models are deployed to evaluate the efficiency of the combined models in classifying hate and offensive comments for Vietnamese.

Table 5 Overview statistics of the two Vietnamese HSD datasets

Dataset	Labels	Percentage (%)	Example
ViHSD	CLEAN	82.71	link đâu th ng kia (English: where is the link, man)
	OFFENSIVE	6.77	vkl. (English: cuss.)
	HATE	10.52	th y đ*t mẹ giả tạo vl =)) (English: the teacher is so f*cking fake =))
HSD-VLSP	CLEAN	91.49	cho xíu nhạc đi (English: some music please)
	OFFENSIVE	5.02	đ*o l y vk nữa đâu (English: no more f*cking married)
	HATE	3.49	th ng giá ch* ch*t (English: f*ck that old man)

Furthermore, compared to BERT-CNN, RoBERTa-CNN, and XLMR-CNN models, our proposed PhoBERT-CNN model provides an insight into the effect of monolingual and multilingual pre-trained language models on this task.

4.3 HSD performance evaluation metric

This section explain the evaluation metrics used in this paper. Accuracy and average macro F1-score are the popular and widely used metric for classification tasks in general and identifying hate and offensive comments in particular [9, 10, 19–24]. However, due to the significantly unbalanced classes in the given datasets, the average macro F1-score, which is the harmonic mean of Precision and Recall, is the most suitable metric for this task [66, 67]. As a result, when evaluating model performance, we opted to utilize the average macro F1-score (%) as the primary metric and the Accuracy (%) to provide additional information. Equations (2)–(5) presents the measures for multi-class classification for multi classes C_i, i $\in$ {1, 2, 3} (denoted by CLEAN, OFFENSIVE and HATE, respectively). Where tp_i are true positive for C_i, and fp_i-false positive, fn_i-false negative, and tn_i-true negative counts respectively. M indices represent macro-averaging.

$$\text{Accuracy} = \frac{\sum_{i=1}^{3} \frac{tp_i + tn_i}{tp_i + fp_i + tn_i + fn_i}}{3} \tag{2}$$

$$\text{Precision}_M = \frac{\sum_{i=1}^{3} \frac{tp_i}{tp_i + fp_i}}{3} \tag{3}$$

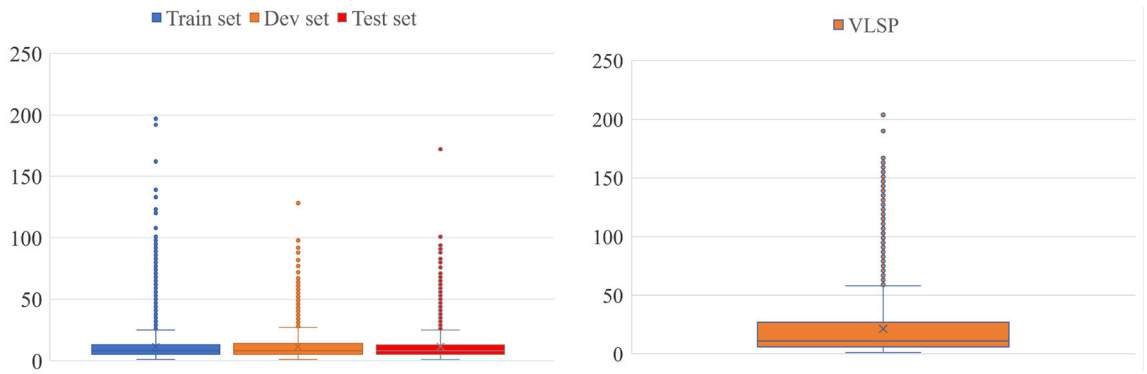
$$\text{Recall}_M = \frac{\sum_{i=1}^{3} \frac{tp_i}{tp_i + fn_i}}{3} \tag{4}$$

$$F1 = 2 * \frac{\text{Precision}_M * \text{Recall}_M}{\text{Precision}_M + \text{Recall}_M} \tag{5}$$

4.4 Vietnamese hate speech detection datasets

We use the ViHSD dataset published by Luu et al. [10] as the primary dataset to build the classification models. The ViHSD dataset [10] consists of 33,400 comments collected from popular social networking sites in Vietnam such as Facebook and Youtube. This dataset is divided into training, development, test sets corresponding to a ratio of 7:1:2.

In addition, to illustrate the solution applicability and efficacy on the social media data domain, we also evaluate our solution on the HSD-VLSP dataset published by Vu et al. [9]. HSD-VLSP is a Vietnamese Hate Speech Detection dataset on social-network comments provided by



(a) ViHSD dataset. **(b)** HSD-VLSP dataset.

Fig. 11 Distribution of the comments length in the two Vietnamese HSD datasets

VSLP 2019 shared-task [9]. This dataset contains 20,345 comments and posts on social networks.

Each comment in both datasets is assigned one of three labels: CLEAN, OFFENSIVE, or HATE. Table 5 shows overview statistics of the datasets. According to the statistics, there is a significant difference in the number of CLEAN comments compared to OFFENSIVE and HATE comments.

The distribution of comment lengths in the ViHSD and HSD-VLSP datasets is depicted in Fig. 11. In the two datasets, ViHSD and HSD-VLSP, we can observe that the average length of the comments is 11.51 and 21.31 words, and the length of the comments is in the range (1; 25) and (1; 58.5), respectively. Besides, we found that comments are often short because users tend to be brief and use many acronyms. As demonstrated in Sect. 3.2.2 and the statistics in Table 3, one of the reasons for the relatively short average length of comments in the two datasets is the inclination of users to overuse acronyms, teen-codes to save time and type faster.

4.5 Experimental settings

As we described in Sect. 3, we experimented on the given datasets with four approaches: traditional machine learning, deep learning, transfer learning, and the combined approach. Following the study of Son et al. [10], all findings from the ViHSD dataset are presented on the test set, while the development set is utilized for hyper-parameter tuning. Furthermore, due to the secrecy of HSD-VLSP, we employ KFoldCrossvalidation (k = 5) for training and testing our models as a solution for the lack of a test set [9, 21, 22].

Sections 4.5.1 and 4.5.2 provide the optimal hyper-parameters that we grid-searched for Machine Learning and Deep Learning models, respectively. On the other hand, we use an Adam optimizer with a fixed learning rate of 2e−5 and a batch size of 64 to fine-tune the hyperparameters of the PhoBERT-CNN model and the other baseline models.

4.5.1 Machine learning approach

In this paper, we implement several machine learning models such as Multinomial Naive Bayes, Logistic Regression, Decision Tree, and Random Forest. In addition, we use the TF-IDF technique with parameter ngram_range is (1,2) for feature extraction. Besides, we also use the class-weigh algorithm to solve the imbalance between three labels, but we do not get any better results.

- *Multinomial Naive Bayes*: We use MutinomialNB with "alpha" = 1.0.
- *Logistic regression*: This model is implemented with the parameters C = 1.0, solver = "lbfgs", maxIter = 20, and regParam = 0.3.
- *Decision tree*: Model parameters include n_estimators = 108, random_state = "None", class_weight = "balanced", max_depth = 17, and min_samples_leaf = 3.
- *Random forest*: We implement a Random Forest Classifier model with numTrees = 200, maxDepth = 10, maxBins = 64.

4.5.2 Deep learning approach

This paper uses these two models: Text-CNN and Bi-LSTM. Otherwise, we also fine-tune those models with novel pre-trained word embeddings, which are Vietnamese word embedding ETNLP [68] and 300 dimensions with character n-grams PhoW2V: the word embedding used for Vietnamese and pre-trained Word2vec syllable [69].

- *Text-CNN*: we set up four conv2D layers with 32 filters at sizes 1, 2, 3, 5 and used softmax for activation. In addition, we set batch size equal to 64, max sequence length is 40, and dropout is 0.4 for this model.
- *Bi-LSTM*: the structure of this model has a bidirectional layer followed by a max-pooling 1D, a dense layer has 50 in size for both activation and softmax activation. We set batch size equal to 64, max sequence length is 68, and dropout is 0.4.

4.5.3 Transfer learning approach

We implement BERT$_{large}$ cased [26], RoBERTa$_{large}$ [27], XLM-R$_{large}$ [28], and PhoBERT$_{large}$ [37] for this approach. They runs with their max sequence length is 60, batch size is 64, learning rate is at 2e−5, accumulation steps are 5, learning rate decay steps are 70.

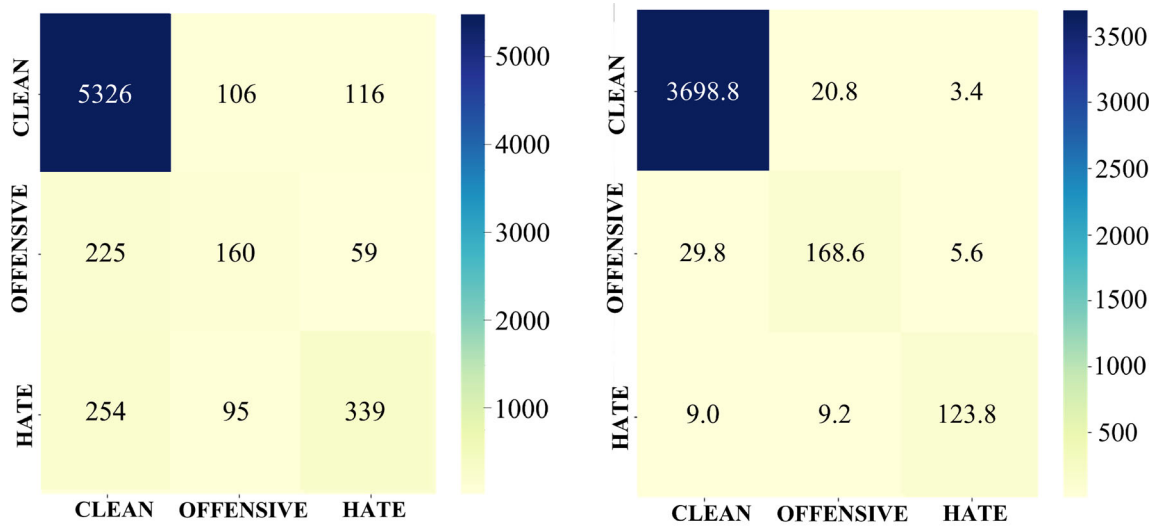
4.5.4 Proposed approach (PhoBERT-CNN)

In this approach, we combine the PhoBERT$_{\text{large}}$ pre-trained model [37] from HuggingFace with the Text-CNN model. The output of PhoBERT$_{\text{large}}$ pre-trained is used as embedding input for the Text-CNN.

- The PhoBERT$_{\text{large}}$ pre-trained is initialized with a max length is 20.
- The Text-CNN is built with four layers of conv1D with filter size is 32 and size 1, 2, 3, 5, respectively.

Table 6 Evaluation results on the two Vietnamese HSD datasets

Models	ViHSD		HSD-VLSP	
	F1-score	Accuracy	F1-score	Accuracy
Multinomial Naive Bayes	50.33	85.23	63.06	92.45
Logistic regression	56.77	86.61	66.41	94.29
Decision tree	55.68	83.38	60.75	91.84
Random forest	54.35	85.45	68.46	95.06
Text-CNN + *fastText*	61.67	86.98	85.76	97.14
Text-CNN + $PhoW2V_{syllable}$	62.49	86.89	86.52	97.22
Text-CNN + $PhoW2V_{word}$	63.01	86.11	85.36	97.11
Bi-LSTM + *fastText*	60.80	85.85	86.21	96.40
Bi-LSTM + $PhoW2V_{syllable}$	60.61	86.35	86.06	97.12
Bi-LSTM + $PhoW2V_{word}$	62.66	85.99	84.04	96.79
BERT	60.29	84.52	85.41	96.19
RoBERTa	61.49	83.04	85.79	96.95
XLM-R	62.38	83.62	86.57	97.15
PhoBERT	63.51	87.13	86.68	97.58
BERT-CNN	61.26	85.90	86.37	96.17
RoBERTa-CNN	62.47	84.54	86.48	96.38
XLMR-CNN	63.34	85.48	88.53	96.92
PhoBERT-CNN	64.43	87.17	90.89	98.26



(a) ViHSD dataset. (b) HSD-VLSP dataset.

Fig. 12 Confusion matrix of our proposed system for Vietnamese HSD

4.6 Analysis and discussion of experimental results

4.6.1 Verifying the performance of the proposed PhoBERT-CNN model

Table 6 shows our results from the experiments conducted. Experiment results show that the PhoBERT-CNN model outperforms traditional machine learning models on two benchmark datasets, ViHSD and HSD-VLSP, respectively, by an F1-score of 7.51 ± 6.59% and 16.25 ± 13.89%. With the deep learning approach, the Text-CNN model outperforms the Bi-LSTM model for the short text classification tasks in general and the HSD problem in particular. Among our single models, PhoBERT archives the highest results on the ViHSD [10] and HSD-VLSP [9] datasets. PhoBERT can perform parallel computations for words, reduce vanishing gradients, and help the model learn better. Our combined

Table 7 Several examples of classification error on the given datasets

Comment	Label	Prediction
Đừng c bi n minh =)))) choi lon	CLEAN	HATE
(English: Do not try to make excuses, play big)		
L m ti n má chả có n i ý thức của loái người :))	CLEAN	HATE
(English: Money can not buy human consciousness :)))		
con náy h t thu c chữa r i	HATE	CLEAN
(English: I am done with this dumb ass)		
Nham	OFFENSIVE	HATE

PhoBERT-CNN model outperforms the baseline models on the ViHSD dataset by 7.51 ± 6.59% in macro F1-score and 2.16 ± 1.97% in Accuracy, respectively. Besides, the proposed approach also demonstrates its efficacy in the Vietnamese social network data domain. On the HSD-VLSP dataset, PhoBERT-CNN achieves the best results, with a macro F1-score of 90.89% and an Accuracy of 98.26%.

On the other hand, the monolingual pre-trained language model for Vietnamese, particularly PhoBERT, outperforms the multilingual models on the task of Vietnamese HSD. Furthermore, combining the BERT and its variants such as PhoBERT, RoBERTa, and XLM-R with the CNN improves their performance by up to 0.98% and 4.21% macro F1-score in the two datasets ViHSD and HSD-VLSP, respectively.

4.6.2 Error analysis and discussion

The confusion matrix of our best-performance model, PhoBERT-CNN, is used for error analysis to analyze the errors encountered in our system. The confusion matrices of our best model when making predictions on the test set are shown in Fig. 12. As a result of the data imbalance, we observe that ability of our system to predict on the CLEAN label is better than the OFFENSIVE and HATE labels.

There are still some comments with misclassification in the dataset due to the ambiguity in identifying the labels. As

Table 8 The results compare the performance of two datasets, ViHSD and HSD-VLSP, for data augmentation techniques

Model	ViHSD				HSD-VLSP			
	F1-score		Accuracy		F1-score		Accuracy	
	W/o	W	W/o	W	W/o	W	W/o	W
Multinomial Naive Bayes	50.33	59.67 (↑9.34)	85.23	84.95 (↓0.28)	63.06	87.93 (↑24.87)	92.45	89.95 (↓2.50)
Logistic Regression	56.77	61.37 (↑4.60)	86.61	88.33 (↑1.72)	66.41	91.91 (↑25.50)	94.29	93.65 (↓0.64)
Decision Tree	55.68	59.66 (↑3.98)	83.38	83.77 (↑0.39)	60.75	89.85 (↑29.10)	91.84	91.54 (↓0.30)
Random Forest	54.35	61.89 (↑7.54)	85.45	86.58 (↑1.13)	68.46	94.01 (↑25.55)	95.06	95.25 (↑0.19)
Text-CNN + *fastText*	61.67	62.54 (↑0.87)	86.98	86.07 (↓0.91)	85.76	98.16 (↑12.40)	97.14	98.29 (↑1.15)
Text-CNN + $PhoW2V_{syllable}$	62.49	63.57 (↑1.08)	86.89	85.14 (↓1.75)	86.52	98.03 (↑11.51)	97.22	98.16 (↑0.94)
Text-CNN + $PhoW2V_{word}$	63.01	63.53 (↑0.52)	86.11	86.06 (↓0.05)	85.36	98.19 (↑12.83)	97.11	98.23 (↑1.12)
Bi-LSTM + *fastText*	60.80	62.85 (↑2.05)	85.85	86.99 (↑1.14)	86.21	97.56 (↑11.35)	96.40	97.92 (↑1.52)
Bi-LSTM + $PhoW2V_{syllable}$	60.61	62.74 (↑2.13)	86.35	87.73 (↑1.38)	86.06	97.47 (↑11.41)	97.12	97.73 (↑0.61)
Bi-LSTM + $PhoW2V_{word}$	62.66	63.68 (↑1.02)	85.99	86.57 (↑0.58)	84.04	97.56 (↑13.52)	96.79	97.74 (↑0.95)
BERT	60.29	63.04 (↑2.75)	84.52	81.39 (↓3.13)	85.41	96.67 (↑11.26)	96.19	98.01 (↑1.82)
RoBERTa	61.49	60.39 (↑0.99)	83.04	83.87 (↑0.83)	85.79	94.44 (↑8.65)	96.95	97.11 (↑0.16)
XLM-R	62.38	64.89 (↑2.51)	83.62	82.07 (↓1.55)	86.57	95.50 (↑8.93)	97.15	97.48 (↑0.33)
PhoBERT	63.51	65.07 (↑1.56)	87.13	89.59 (↑2.46)	86.68	97.01 (↑10.33)	97.58	98.55 (↑0.97)
BERT-CNN	61.26	62.99 (↑3.73)	85.90	75.57 (↓0.33)	86.37	96.54 (↑10.17)	96.17	97.98 (↑1.81)
RoBERTa-CNN	62.47	64.02 (↑1.55)	84.54	88.06 (↑3.52)	86.48	95.47 (↑8.99)	96.38	98.01 (↑1.63)
XLMR-CNN	63.34	67.29 (↑3.95)	85.48	80.57 (↓4.91)	88.53	97.85 (↑9.32)	96.92	98.04 (↑1.12)
PhoBERT-CNN	64.43	67.46 (↑3.03)	87.17	87.76 (↑6.41)	90.89	98.45 (↑7.56)	98.26	98.59 (↑0.33)

"W" and "W/o" denote that the results are evaluated with data augmentation techniques and without data augmentation techniques

Table 9 The comparison with previous pre-processing techniques on ViHSD test set and HSD-VLSP dataset

Model	ViHSD				HSD-VLSP			
	F1-score		Accuracy		F1-score		Accuracy	
	LT	OT	LT	OT	HT	OT	HT	OT
Multinomial Naive Bayes	57.70	59.67 (↑1.97)	85.01	84.95 (↓0.06)	61.50	87.93 (↑26.43)	91.54	89.95 (↓1.59)
Logistic regression	54.53	61.37 (↑6.84)	86.27	88.33 (↑2.06)	63.98	91.91 (↑27.93)	94.01	93.65 (↓0.36)
Decision tree	55.48	59.66 (↑4.18)	83.48	83.77 (↑0.29)	59.08	89.85 (↑30.77)	91.10	91.54 (↑1.44)
Random forest	54.01	61.89 (↑7.88)	85.61	86.58 (↑0.97)	66.26	94.01 (↑27.75)	94.21	95.25 (↑1.04)
Text-CNN + Fasttext	61.95	62.54 (↑0.59)	86.89	86.07 (↓0.82)	85.02	98.16 (↑13.14)	97.06	98.29 (↑1.23)
Text-CNN + Pho2Vec_syllable	59.25	63.57 (↑4.32)	86.03	85.14 (↓0.89)	84.99	98.03 (↑13.04)	97.01	98.16 (↑1.15)
Text-CNN + Pho2Vec_word	60.01	63.53 (↑3.52)	86.44	86.06 (↑0.38)	84.28	98.19 (↑13.91)	96.24	98.23 (↑1.99)
Bi-LSTM + Fasttext	60.37	62.85 (↑2.48)	85.18	86.99 (↑1.81)	85.33	97.56 (↑12.23)	93.86	97.92 (↑4.06)
Bi-LSTM + Pho2Vec_syllable	60.27	62.74 (↑2.47)	84.42	87.73 (↑3.31)	82.12	97.47 (↑15.35)	95.02	97.73 (↑2.71)
Bi-LSTM + Pho2Vec_word	61.09	63.68 (↑2.59)	85.04	86.57 (↑1.53)	82.18	97.56 (↑15.38)	94.57	97.74 (↑3.17)
BERT	53.85	63.04 (↑9.19)	82.47	81.39 (↓1.08)	65.11	96.67 (↑31.56)	96.79	98.01 (↑1.22)
RoBERTa	55.34	60.39 (↑5.05)	83.54	83.87 (↑0.33)	84.98	94.44 (↑9.46)	96.01	97.11 (↑1.10)
XLM-R	61.28	64.89 (↑3.61)	86.12	82.07 (↓4.05)	85.34	95.50 (↑10.16)	96.88	97.48 (↑0.60)
PhoBERT	60.58	65.07 (↑4.49)	86.84	89.59 (↑2.75)	86.02	97.01 (↑10.99)	96.58	98.55 (↑1.97)
BERT-CNN	59.81	64.02 (↑4.21)	84.57	88.06 (↑3.49)	85.01	95.47 (↑10.46)	95.17	98.01 (↑2.84)
RoBERTa-CNN	58.28	62.99 (↑4.71)	84.01	75.57 (↓8.44)	85.11	96.54 (↑11.43)	95.55	97.98 (↑2.43)
XLMR-CNN	62.39	67.29 (↑4.90)	84.68	80.57 (↓4.11)	87.34	97.85 (↑10.51)	95.08	98.04 (↑2.96)
PhoBERT-CNN	62.66	67.46 (↑4.80)	87.07	87.76 (↑0.69)	88.75	98.45 (↑9.70)	98.07	98.59 (↑0.52)

"LT," "HT," and "OT" refer to Luu et al. [10], Huynh et al. [21], and our proposed pre-processing techniques, respectively

described in Sect. 3.4, the combination of the PhoBERT and the CNN contributes to improvement of the performance of the classifier by extracting the features related to the keywords representing the main idea of the sentence. However, CNN's extraction and learning of these keywords are sometimes too sensitive and subjective, leading to confusion in detecting the corresponding label of the comments. We can see that many misclassified comments are affected by the decision keywords, such as in the label CLEAN but are misclassified with the topic HATE due to the keyword "lon$_{big/pussy}$", "nham$_{miss/bullshit}$", "Đm$_{fuck/fate}$". In addition, the HATE label and OFFENSIVE label is often confused with the CLEAN label because they contain racist or insinuating content that makes them challenging to predict [10, 18].

Table 10 The comparison with previous studies on ViHSD dataset

Model	F1-score	Accuracy
GRU + fastText [10]	60.47	85.41
Text-CNN + fastText [10]	61.11	86.69
XLM-R [10]	61.28	86.12
DistilBERT [10]	62.42	86.22
BERT [10]	62.69	86.88
Our approach (PhoBERT-CNN)	67.46	87.76

Table 11 The comparison with previous studies on HSD-VLSP dataset

Model	F1-score
Bi-LSTM* [19]	56.28
DCNN, Text-CNN, LSTM, LSTMCNN, SARNN* [23]	58.45
Logistic regression + Random Forest + Extra Tree* [24]	58.88
Logistic regression* [20]	61.97
Text-CNN [22]	83.04
CNN + Bi-LSTM + LSTM [21]	86.96
Our approach (PhoBERT-CNN)	98.45

*The result is evaluated on a test set of the VLSP shared task 2019. Others use K-fold cross-validation to evaluate the model ($k = 5$) following the study

4.6.3 Augumentation data results

One of the challenging problems in our task, as described in Sect. 4.4, is data imbalance, which has a negative impact on

Table 12 Ablation test on our proposed approach

OP	DA	PB	TC	ViHSD		HSD-VLSP	
				F1-score	Accuracy	F1-score	Accuracy
✓	✓	✓	✓	67.46	87.76	98.45	98.59
✓	✓	✓	✗	65.07(↓2.39)	87.59 (↓0.17)	97.01(↓1.44)	98.55 (↓0.04)
✓	✓	✗	✓	63.54 (↓3.92)	86.06 (↓1.70)	98.19 (↓0.26)	98.23 (↓0.36)
✓	✗	✓	✓	64.43 (↓3.03)	87.17 (↓0.59)	90.89 (↓7.56)	98.26 (↓0.33)
✓	✗	✓	✗	63.51 (↓3.95)	87.13 (↓0.63)	86.68 (↓11.77)	97.58 (↓1.01)
✓	✗	✗	✓	63.01 (↓4.45)	86.11 (↓1.65)	85.36 (↓13.09)	97.11 (↓1.48)
✗	✓	✓	✓	62.91 (↓4.55)	86.41 (↓1.35)	88.75 (↓9.70)	98.07 (↓0.52)
✗	✓	✓	✗	62.08 (↓5.38)	86.38 (↓1.38)	86.02 (↓12.43)	96.88 (↓1.71)
✗	✓	✗	✓	61.86 (↓5.60)	85.71 (↓2.05)	85.99 (↓12.46)	96.24 (↓2.35)
✗	✗	✓	✓	61.45 (↓6.01)	85.25 (↓2.51)	84.61 (↓13.84)	97.85 (↓0.74)
✗	✗	✓	✗	58.87 (↓8.59)	86.34 (↓1.42)	84.52 (↓13.93)	96.65 (↓1.94)
✗	✗	✗	✓	56.03 (↓11.43)	86.54 (↓1.22)	83.46 (↓14.99)	96.33 (↓2.26)

"OP", "DA", "PB", and "TC" denote the use of our proposed pre-processing, the data augmentation techniques, the pre-trained $\text{PhoBERT}_{\text{large}}$ model, and the Text-CNN model, respectively

classification model performance. As a result, inspired by the work of Wei and Zou [46], we intend to apply EDA techniques to the ViHSD dataset and the HSD-VLSP dataset in this paper to deal with imbalanced data and improve the performance of classification models.

We used the data divided into training, development, and test sets by Luu et al. [10] for the ViHSD dataset. The training set was used to implement EDA techniques, the development set to fine-tune classifier hyper-parameters, and the test set to evaluate our models.

For training and testing our models on the HSD-VLSP dataset, we employed five-fold cross-validation. Following the same approach as in the previous study [22, 44], we preserve the test set and augment the training set with EDA techniques for each fold.

Table 8 presents the results of a comparison of model performance with and without data augmentation techniques. The results indicate that using data augmentation approaches improves models performance by up to 9.34% and 24.87% macro F1-score in the two datasets, ViHSD and HSD-VLSP, respectively. Our proposed PhoBERT-CNN model, on the other hand, outperforms the baseline models 4.93 ± 4.41% and 16.76 ± 8.11% macro F1-score. According to the results, we conclude that using appropriate data augmentation techniques can significantly improve the performance of models. However, in some cases, data augmentation techniques could decrease model accuracy. As a result, it is critical to investigate if applying data augmentation techniques to a certain situation is appropriate.

4.6.4 Comparison with previous studies

We conducted experiments that compare the performance of our proposed data pre-processing techniques against the previous pre-processing, including comparing it against the Luu et al. [10] and Huynh et al. [21] studies. We choose to compare with the studies of Luu et al., and Huynh et al. [21] because these are the works that implement effective pre-processing techniques and achieve more positive results than previous studies on the same dataset and evaluation measure [10, 21]. The experimental results are presented in Table 9. The results show that our proposed pre-processing techniques significantly improve the performance of the PhoBERT-CNN model (increase by 4.80% and 9.70% macro F1-score on two datasets, ViHSD and HSD-VLSP). Moreover, our proposed pre-processing techniques improve the performance of the baseline models 2.82 ± 2.23% and 20.51 ± 11.05% macro F1-score on two datasets, ViHSD and HSD-VLSP, respectively. As a result, we conclude that the proposed pre-processing techniques are efficient in Vietnamese hate speech detection on social media.

Our approach outperforms previous studies on the ViHSD and HSD-VLSP datasets. The same evaluation metrics as previous studies are used to make fair comparisons. We utilize the average macro F1 (%), Accuracy (%) for ViHSD dataset [10] and the average macro F1 score (%) for HSD-VLSP dataset [9]. The Tables 10 and 11 show the best results we achieved compared to previous studies. Our combined PhoBERT-CNN model outperforms baseline models of Luu et al. [10] on the ViHSD dataset by 5.88 ± 1.11% macro F1-score and 1.615 ± 0.735% Accuracy, respectively. Furthermore, PhoBERT-CNN also

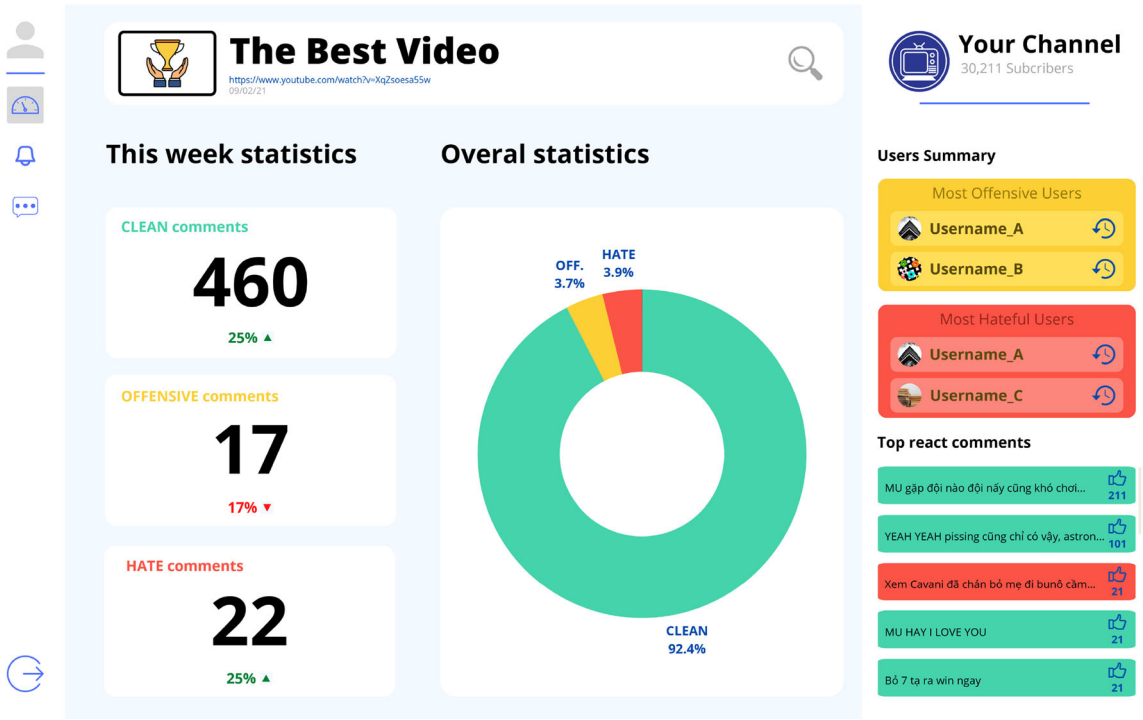


Fig. 13 The interface of the HSD application with streaming data

achieves the best results, with a macro F1-score of 98.45% on the HSD-VLSP dataset.

4.6.5 Ablation analysis of proposed method

Our proposed approach is significantly more straightforward and efficient than most existing Vietnamese hate speech detection approaches (see Sect. 4.6.4). An ablation analysis was performed on the proposed approach to demonstrate the efficacy and alignment of the modules. Table 12 shows that each module contributes to the overall performance of our approach. Without the pre-trained PhoBERT model, our method only achieved 63.54% and 98.19% macro F1-score on the ViHSD and HSD-VLSP datasets, respectively. Likewise, the inclusion of a combined model contributes to a significant improvement in approach performance up to 11.77% in macro F1-score.

We have also compared our proposed pre-processing techniques with the pre-processing techniques of previous works [21]. The Easy Data Augmentation (EDA) techniques [46] was chosen since it is one of the novel solutions and has been shown to be successful when applied to sentiment analysis tasks, including the Vietnamese HSD task [44]. The results show that the EDA techniques significantly improve the performance of the PhoBERT-CNN model (increase by 6.01% and 13.84% macro F1-score in the two datasets, ViHSD and HSD-VLSP). These results verify the need for data augmentation for the Vietnamese HSD problem.

Moreover, removing pre-processing and pre-trained PhoBERT or Text-CNN models reduces performance dramatically. The results demonstrate the importance of data pre-processing in general, as well as the effectiveness of the pre-processing techniques we applied to identify Vietnamese HSD in particular. Combined models, especially pre-trained models combined with deep learning networks, could yield promising outcomes for improving performance in further study. As a result, we conclude that all proposed modules are crucial in Vietnamese hate speech detection on social media.

4.6.6 Hate speech detection application with streaming data

This section runs a pilot experiment on a system with a NVIDIA GEFORCE GTX 1650Ti GPU, an Intel i7-9750H CPU, and 16 GB of RAM. This experiment is conducted to analyze the effectiveness of our proposed system when deployed to handle streaming for a social networking platform in practice. We chose Youtube to conduct the pilot experiment because it is one of the largest social networking sites in Vietnam currently [70]. Besides, Youtube is the primary data source for constructing both the ViHSD [10] and HSD-VLSP [9] datasets. The successful construction and application of the system to the Youtube platform

demonstrates the value of our study. Furthermore, our proposed system lays the groundwork for future research into HSD application systems for social networks.

At the online stage, the results are evaluated by verifying the prediction results returned by the system. Firstly, we randomly collect 500 comments with real-time access from Youtube via Youtube Data API. The collected comments will be stored in JSON format. Collected comments have many attributes and mainly include user-based and comment-based properties. However, we only focus on the comment-based description to classify the polarity labels that the comment conveys.

Next, we hired three annotators with extensive experience constructing NLP datasets for Vietnamese, particularly ViHSD datasets, to annotate the collected comments. Three annotators independently label 500 comments with an inter-annotator agreement score of Cohen Kappa [71] at $K = 0.64$. Figure 13 shows an overview of statistics and the user interface of our application.

Performance of our Hate speech detection application is evaluated through accuracy and latency. Regarding accuracy, the F1-score macro and Accuracy metrics were used to evaluate the system's accuracy. With 500 test data points, our proposed system achieves an F1-score and Accuracy of 58.19% and 82.02%, respectively. Latency is measured as the average response time of our system for processing and classifying a batch of data collected during a session. Our system achieve a latency of 1.56 s. Note that Apache Spark Structured Streaming 3.1.1[2] [50] was used in our system. This version includes Continuity Processing for Structured Streaming, which can provide low-latency responses on the order of miliseconds. However, due to limited hardware resources, our system can only achieve a modest latency of 1.56 s. Having said that, this level of latency certainly satisfy the requirement of an application of real-time hate speech detection dealt with in this study. Furthermore, our system could serve as the foundation for further development in the field.

5 Conclusion and future works

This research proposes a novel state-of-the-art solution to the Vietnamese HSD task. We introduced a new data processing process with two phases to clean the given dataset well. The results show that our proposed pre-processing techniques significantly improve the performance of the PhoBERT-CNN model. Moreover, our proposed pre-processing techniques improve the performance of the baseline models on ViHSD and HSD-VLSP datasets. We were also successful in addressing the problems of data imbalance by applying the EDA techniques. Meanwhile, an efficient and straightforward approach for Vietnamese hate speech detection based on the combined model PhoBERT-CNN was proposed. The proposed combined model outperformed the baseline models and previous studies on the same dataset and evaluation measure. We achieved F1-score results on the ViHSD and HSD-VLSP datasets of 67.46% and 98.45%, respectively. Furthermore, we have successfully built a real-time hate speech detection system for Vietnamese using Spark streaming. The results obtained are pretty optimistic and reliable for solving the task, helping to reduce the occurrence of hate or offensive comments, and building a healthy and safe environment.

Inspired by the success of Mozafari et al. [72], we intend to implement more Vietnamese pre-trained language models to find a better model that achieves better performance in the Hate Speech Detection task. Moreover, our research lays the groundwork for future research in areas such as: (1) detecting multiple aspects and human rationales of hate speech [73]; (2) detecting hate and offensive spans at the word and phrase levels [74].

Acknowledgments This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mohan S, Guha A, Harris M, Popowich F, Schuster A, Priebe C (2017) The impact of toxic language on the health of reddit communities. In: Canadian conference on artificial intelligence. Springer, pp 51–56
2. Abu-Ghazaleh S, Hassona Y, Hattar S (2018) Dental trauma in social media-analysis of facebook content and public engagement. Dent Traumatol 34(6):394–400
3. Statista: Global number of hate speech-containing content removed by Facebook from 4th quarter 2017 to 2nd quarter 2021 (2018). https://www.statista.com/statistics/1013804/facebook-hate-speech- content-deletion-quarter
4. Seetharaman D (2018) Facebook throws more money at wiping out hate speech and bad actors. https://www.wsj.com/articles/facebook-throws-more-cash-at-tough-problem-stamping-out-bad-content-15263932
5. Microsoft: Global number of hate speech-containing content removed by Facebook from 4th quarter 2017 to 2nd quarter 2021 (2020). https://www.microsoft.com/en-us/online-safety/digital-civility
6. Keane TM, Fisher LM, Krinsley KE, Niles BL (1994) Posttraumatic stress disorder. Springer, Berlin, pp 237–260
7. Malmasi S, Zampieri M (2017) Detecting hate speech in social media. In: Proceedings of the international conference recent advances in natural language processing. INCOMA Ltd., Varna, pp 467–472. https://doi.org/10.26615/978-954-452-049-6_062

[2] https://spark.apache.org/docs/3.1.1.

8. Schmidt A, Wiegand M (2017) A survey on hate speech detection using natural language processing. In: Proceedings of the fifth international workshop on natural language processing for social media, pp 1–10
9. Vu X-S, Vu T, Tran M-V, Le-Cong T, Nguyen H (2020) HSD shared task in VLSP campaign 2019: hate speech detection for social good. arXiv preprint. arXiv:2007.06493
10. Luu ST, Nguyen KV, Nguyen NL-T (2021) A large-scale dataset for hate speech detection on Vietnamese social media texts. In: Fujita H, Selamat A, Lin JC-W, Ali M (eds) Advances and trends in artificial intelligence. Artificial intelligence practices. Springer, Cham, pp 415–426
11. Naseem U, Razzak I, Eklund PW (2021) A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimed Tools Appl 80 (28):35239–35266
12. Nguyen KP-Q, Van Nguyen K (2020) Exploiting Vietnamese social media characteristics for textual emotion recognition in Vietnamese. In: International conference on Asian language processing (IALP). IEEE, pp 276–281
13. Vu T, Nguyen DQ, Nguyen DQ, Dras M, Johnson M (2018) VnCoreNLP: a Vietnamese natural language processing toolkit. In: Proceedings of the 2018 conference of the North American Chapter of the Association for computational linguistics: demonstrations. Association for Computational Linguistics, New Orleans, pp 56–60. https://doi.org/10.18653/v1/N18-5012
14. Fortuna P, Nunes S (2018) A survey on automatic detection of hate speech in text. ACM Comput Surv (CSUR) 51(4):1–30
15. Alrehili A (2019) Automatic hate speech detection on social media: a brief survey. In: IEEE/ACS 16th International conference on computer systems and applications (AICCSA). IEEE, pp. 1–6
16. Waseem Z, Hovy D (2016) Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In: Proceedings of the NAACL student research workshop, pp 88–93
17. Chen J, Yan S, Wong K-C (2018) Verbal aggression detection on twitter comments: convolutional neural network for short-text sentiment analysis. Neural Comput Appl 32:10809–10818
18. Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. In: Proceedings of the international AAAI conference on web and social media, vol 11
19. Do HT-T, Huynh HD, Van Nguyen K, Nguyen NL-T, Nguyen AG-T (2019) Hate speech detection on Vietnamese social media text using the bidirectional-lstm model. arXiv preprint. arXiv:1911.03648
20. Huu QP, Trung SN, Pham HA (2019) Automated hate speech detection on Vietnamese social networks. Technical report, EasyChair
21. Huynh HD, Do HT-T, Nguyen KV, Nguyen NT-L (2020) A simple and efficient ensemble classifier combining multiple neural network models on social media datasets in Vietnamese. In: Proceedings of the 34th Pacific Asia conference on language, information and computation. Association for Computational Linguistics, Hanoi, pp 420–429
22. Luu ST, Nguyen HP, Van Nguyen K, Nguyen NL-T (2020) Comparison between traditional machine learning models and neural network models for Vietnamese hate speech detection. In: RIVF international conference on computing and communication technologies (RIVF). IEEE, pp 1–6
23. Nguyen TB, Nguyen QM, Nguyen TH, Pham NP, Nguyen TL, Do QT (2019) Vais hate speech detection system: a deep learning based approach for system combination. arXiv preprint. arXiv:1910.05608
24. Van Thin D, Le LS, Nguyen NL-T (2019) Nlp@ uit: Exploring feature engineer and ensemble model for hate speech detection at vlsp 2019. Training 5:3–51
25. Martins R, Gomes M, Almeida JJ, Novais P, Henriques P (2018) Hate speech classification in social media using emotional analysis. In: 7th Brazilian conference on intelligent systems (BRACIS). IEEE, pp 61–66
26. Devlin J, Chang M-W, Lee K, Toutanova K (2019) 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, vol 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423
27. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint. arXiv:1907.11692
28. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2020) Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th annual meeting of the association for computational linguistics. Association for Computational Linguistics, pp 8440–8451 (Online). https://doi.org/10.18653/v1/2020.acl-main.747
29. Safaya A, Abdullatif M, Yuret D (2020) Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In: Proceedings of the fourteenth workshop on semantic evaluation, pp 2054–2059
30. Liu Y, Liu H, Wong L-P, Lee L-K, Zhang H, Hao T (2020) A hybrid neural network rbert-c based on pre-trained roberta and cnn for user intent classification. In: International conference on neural computing for advanced applications. Springer, pp 306–319
31. Saha D, Paharia N, Chakraborty D, Saha P, Mukherjee A (2021) Hate-alert@DravidianLangTech-EACL2021: ensembling strategies for transformer-based offensive language detection. In: Proceedings of the first workshop on speech and language technologies for Dravidian languages. Association for Computational Linguistics, Kyiv, pp 270–276
32. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
33. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
34. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint. arXiv:1412.3555
35. He C, Chen S, Huang S, Zhang J, Song X (2019) Using convolutional neural network with bert for intent determination. In: International conference on Asian language processing (IALP). IEEE, pp 65–70
36. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, pp 1746–1751. https://doi.org/10.3115/v1/D14-1181
37. Nguyen DQ, Tuan Nguyen A (2020) PhoBERT: pre-trained language models for Vietnamese. In: Findings of the association for computational linguistics: EMNLP 2020. Association for Computational Linguistics, pp 1037–1042 (Online). https://doi.org/10.18653/v1/2020.findings-emnlp.92
38. Nagarajan SM, Gandhi UD (2019) Classifying streaming of twitter data based on sentiment analysis using hybridization. Neural Comput Appl 31(5):1425–1433
39. Zaki ND, Hashim NY, Mohialden YM, Mohammed MA, Sutikno T, Ali AH (2020) A real-time big data sentiment analysis for iraqi tweets using spark streaming. Bull Electric Eng Inform 9(4):1411–1419
40. Burnap P, Williams ML (2015) Cyber hate speech on twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet 7(2):223–242

41. Anagnostou A, Mollas I, Tsoumakas, G (2018) Hatebusters: a web application for actively reporting youtube hate speech. In: IJCAI, pp 5796–5798
42. Bird S (2006) Nltk: the natural language toolkit. In: Proceedings of the COLING/ACL 2006 interactive presentation sessions, pp 69–72
43. Le V-D (2017) Stopwords: Vietnamese. GitHub
44. Luu S, Nguyen K, Nguyen N (2020) Empirical study of text augmentation on social media text in Vietnamese. In: Proceedings of the 34th Pacific Asia conference on language, information and computation. Association for Computational Linguistics, Hanoi, pp 462–470
45. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
46. Wei J, Zou K (2019) EDA: easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, pp 6382–6388. https://doi.org/10.18653/v1/D19-1670
47. Pham-Hong B-T, Chokshi S (2020) PGSG at SemEval-2020 task 12: BERT-LSTM with tweets' pretrained model and noisy student training method. In: Proceedings of the fourteenth workshop on semantic evaluation, pp 2111–2116
48. Li X, Bing L, Zhang W, Lam W (2019) Exploiting BERT for end-to-end aspect-based sentiment analysis. In: Proceedings of the 5th workshop on noisy user-generated text (W-NUT 2019). Association for Computational Linguistics, Hong Kong, pp 34–41. https://doi.org/10.18653/v1/D19-5505
49. Yi R, Hu W (2019) Pre-trained BERT-GRU model for relation extraction. In: Proceedings of the 2019 8th international conference on computing and pattern recognition, pp 453–457
50. Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, Meng X, Rosen J, Venkataraman S, Franklin MJ et al (2016) Apache spark: a unified engine for big data processing. Commun ACM 59(11):56–65
51. Rish I et al (2001) An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop on empirical methods in artificial intelligence, vol 3, pp 41–46
52. Kim S-B, Rim H-C, Yook D, Lim H-S (2002) Effective methods for improving naive Bayes text classifiers. In: Pacific rim international conference on artificial intelligence. Springer, pp 414–423
53. Liu S, Forss T (2014) Combining N-gram based similarity analysis with sentiment analysis in web content classification. In: KDIR, pp 530–537
54. Genkin A, Lewis DD, Madigan D (2007) Large-scale Bayesian logistic regression for text categorization. Technometrics 49 (3):291–304
55. Hosmer DW Jr, Lemeshow S, Sturdivant RX (2013) Applied logistic regression, vol 398. Wiley, Hoboken
56. Pranckevičius T, Marcinkevičius V (2017) Comparison of naive Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. Baltic J Mod Comput 5(2):221
57. Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques. WSEAS Trans Comput 4 (8):966–974
58. Burnap P, Williams ML (2016) Us and them: identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Sci 5:1–15
59. Liaw A, Wiener M et al (2002) Classification and regression by randomforest. R news 2(3):18–22
60. Islam MZ, Liu J, Li J, Liu L, Kang W (2019) A semantics aware random forest for text classification. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 1061–1070
61. Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets. In: Proceedings of the 26th international conference on world wide web companion, pp 759–760
62. Medsker L, Jain LC (1999) Recurrent neural networks: design and applications. CRC Press, Boca Raton
63. Tenney I, Das D, Pavlick E (2019) BERT rediscovers the classical NLP pipeline. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, pp 4593–4601. https://doi.org/10.18653/v1/P19-1452
64. Michel P, Levy O, Neubig G (2019) Are sixteen heads really better than one? In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems, vol 32. Curran Associates, Inc., Red Hook
65. Rogers A, Kovaleva O, Rumshisky A (2020) A primer in bertology: what we know about how bert works. Trans Assoc Comput Linguist 8:842–866
66. Sigurbergsson GI, Derczynski L (2019) Offensive language and hate speech detection for Danish. arXiv preprint. arXiv:1908.04531
67. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1):1–13
68. Vu Xuan S, Vu T, Tran S, Jiang L (2019) ETNLP: a visual-aided systematic approach to select pre-trained embeddings for a downstream task. In: Proceedings of the international conference on recent advances in natural language processing (RANLP 2019). INCOMA Ltd., Varna, pp 1285–1294. https://doi.org/10.26615/978-954-452-056-4_147
69. Nguyen AT, Dao MH, Nguyen DQ (2020) A pilot study of text-to-SQL semantic parsing for Vietnamese. In: Findings of the association for computational linguistics: EMNLP 2020, pp 4079–4085
70. Datareportal: Digital 2021: Vietnam (2021). https://datareportal.com/reports/digital-2021-vietnam
71. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46
72. Mozafari M, Farahbakhsh R, Crespi N (2019) A bert-based transfer learning approach for hate speech detection in online social media. In: International conference on complex networks and their applications. Springer, pp 928–940
73. Mathew B, Saha P, Yimam SM, Biemann C, Goyal P, Mukherjee A (2021) Hatexplain: a benchmark dataset for explainable hate speech detection. Proc AAAI Conf Artif Intell 35(17):14867–14875
74. Pavlopoulos J, Sorensen J, Laugier L, Androutsopoulos I (2021) Semeval-2021 task 5: toxic spans detection. In: Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021), pp 59–69