Big Data-powered Vietnamese Hate Speech Detection:A Kafka - Deep Learning Approach

Chi-Thanh Dang[1,2], Thuy-Hong T. Dang[1,2], Van-Nguyen Dinh[1,2], Trong-Hop Do[1,2]

[1] Faculty of Information Science and Engineering, University of Information Technology, Vietnam National University,Ho Chi Minh City, Vietnam.
[2] 20520761@gm.uit.edu.vn
20520523@gm.uit.edu.vn
20520657@gm.uit.edu.vn
hopdt@.uit.edu.vn

In this research, we have developed a Kafka system to detect hate speech on live videos streamed on the YouTube platform. The system consists of two main components: offline and online. In the first component, we conducted experiments with various Machine Learning and Deep Learning models using two datasets to identify the most effective model. In the online component, comments from the chat box of live YouTube videos are crawled, and the model predicts whether each comment is classified as hate speech or clean. Subsequently, we label and evaluate these comments to improve the model's performance, especially when it encounters misclassifications.

Keywords: Hate Speech Detection, Kafka Streaming, Machine Learning, Deep Learning.

1 Introduction

Nowadays, the internet has become an indispensable part of human life, providing us with diverse and abundant sources of information. However, it cannot be denied that 80% of social media users in Vietnam have been victims of or encountered cases of hate speech (according to figures published by VTV in 2021).

To partially mitigate this issue, we have developed a Kafka system to predict whether a comment is hate speech or not. Firstly, we fine-tuned two datasets, namely ViHSD and VLSP, to make them cleaner and more suitable for the problem. Subsequently, we experimented with various Machine Learning and Deep Learning models, evaluated their performance, and selected the best model. In the next stage, we crawled comments from the chat box of live YouTube videos. We then used Kafka to stream the data to the best model and make predictions. After the comments were predicted, we checked the accuracy and F1 score to further optimize the model's predictions.

The remaining part of this paper, Section 2, addresses related works. The Real-time Hate Speech Detection System, including both offline and online components, will be presented in Section 3. Section 4 covers the results of offline and online learning, while Section 5 discusses about error analysis as well as the system's speed. Lastly, there will be a conclusion and future Work in section 6.

2 Related Work

A proposed Bi-GRU-LSTM-CNN [1] model utilizing the pre-trained FastText model achieved relatively high performance in predicting on the VLSP 2019 dataset. Subsequently, another research group from UIT introduced a Bidirectional-LSTM [2] model combined with FastText word embeddings and baomoi.vn.model.txt [3], yielding even higher results compared to the Bi-GRU-LSTM-CNN model on the same dataset. Regarding the ViHSD dataset, conducted experiments using both Machine Learning Models with TF-IDF and Transfer Learning Models [4]. Moreover, not only experimented with various different models but also employed Kafka and Spark to create a System for Streaming Social Media Data [5].

3 Real-time Hate Speech Detection System

The proposed real-time Hate Speech Detection system consists of two main components: a model trained on a set of two labeled Vietnamese Hate Speech Detection Datasets (offline) and a real-time Hate Speech Detection system (online) as shown in Figure 1.

3.1 Offline Learning

In this section, we propose a simple approach to the problem of Vietnamese Hate Speech Detection. We focus on preprocessing and training models on two datasets to find the best model for the Online Prediction Task. Offline Learning pipeline is shown in Figue 2 below.

3.1.1 Dataset We use two included datasets, ViHSD [6] and HSD-VLSP [7] to train the models. The ViHSD dataset includes 33,400 social media comments, independently labeled by 4 people with a Cohen Kappa consensus of K = 0.52. The HSD-VLSP dataset is a dataset containing Facebook social media posts and comments published at the VLSP2019 contest, the dataset includes 25,431 comments. Each data line of the training set is assigned one of three labels CLEAN, OFFENSIVE, or HATE.
We found that the original two datasets had a huge difference in the number of comments labeled CLEAN compared to comments labeled OFFENSIVE and HATE. However, with the input data of online data from playing videos, the number of comments containing hate speech is very large, so we have combined two labels OFFENSIVE and HATE into a label of HATE so that the model can be trained and evaluated better. Moreover, both datasets suffer from label imbalance, and we want to conduct experiments to compare the model performance on datasets with balanced and imbalanced labels. Therefore, we will remove a

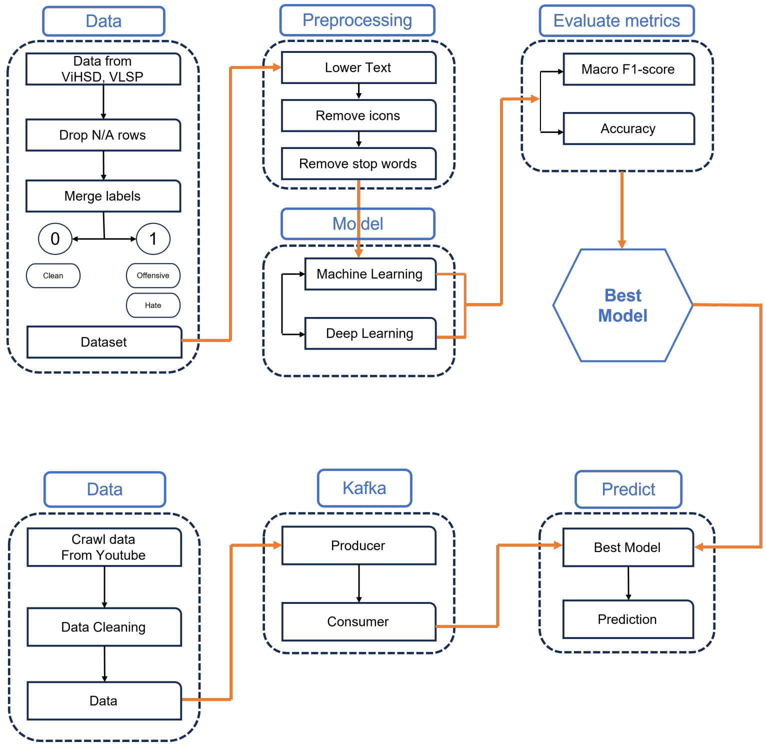

Fig. 1. Architecture of the real-time Hate Speech Detection system.

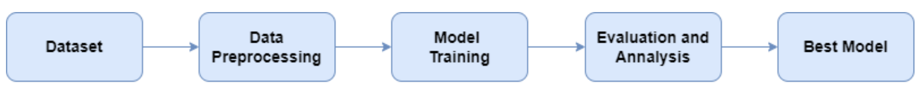

Fig. 2. Offline Learning Pipeline

certain number of instances with the CLEAN label from the ViHSD dataset. This task aims to detect whether a comment on a social network is HATE or CLEAN. Formally, the task is described as follows:

Input: Comments in Vietnamese on social networking sites.
Output: One of two different labels is predicted:

- HATE comments contain abusive language, often causing intended to offend individuals or groups and may include hate speech, insults and insults.
- CLEAN comments are normal comments. Those are normal conversations, emotions. It does not contain offensive language or hate speech.

In addition, the number of duplicate comments and the number of unlabeled comments is quite large, so we have removed the duplicate and unlabeled comments. Finally, we obtain the datasets with the ratio of 2 labels CLEAN and

Table 1. Overview statistics of two datasets: ViHSD and HSD-VLSP

Datasets	Labels	Percentages Before	Percentages After
ViHSD	CLEAN	82.71%	55.49%
	HATE	10.52%	45.51%
	OFFENSIVE	6.77%	-
HSD-VLSP	CLEAN	91.49%	92%
	HATE	3.49%	8%
	OFFENSIVE	5.02%	-

OFFENSIVE balanced for training the model more effectively. The overview statistics of two datasets before and after cleaning show in Table 1 below.

3.1.2 Data Preprocessing Data preprocessing is very important in any social network data analysis because it directly affects the accuracy of the predicted label. Social media data is the most complex textual data source because it includes many links, hashtags, special symbols, emojis, and especially teencode (social networking language) etc. Therefore, data collected from social networks needs to be preprocessed by several methods.
Lowercase Sentences: Converts the letters in the text including uppercase and lowercase letters to lowercase. For example, "Cái này hay nè" is converted to "cái này hay nè".
Normalize Unicode: We also see a lot of Vietnamese words in the dataset that are the same, but python sees them as separate because of its Unicode. The reason why is that there are many Unicode Transformation Formats (UTF) such as UTF-8, UTF-16, UTF-32, and so on are used widely, but our choice is normalizing to UTF-8.
Remove redundant characters and De-Teencode: In the language of social networks, there are many words that have been altered, making it difficult for models to distinguish and assume they are the same. So we removed the redundant characters and normalized them. For example, "kaka" or "kkkkkk" or "kkkkak" both mean laugh "haha" so it is normalized to "haha". Or words like "cn traiiiiiii" are normalized to "con trai" **Noise removal:** This will remove noise in the comment, remove unnecessary characters:

- Remove the characters @User (referring to the user) in the text.
- Remove the # character from the Hashtag in the text.
- Delete URL, Email. In this step, we have removed the links attached to the text.
- Remove special symbols, numbers and remove extra spaces in the text.
- Convert emoticons (emoji) into characters.

Word tokenzie: The input sentence is splitting into words or meaningful word phrases. In order to do this, we use Word Segmenter of VnCoreNLP [8].
Remove stopwords: We also remove stopwords from the comments be- cause

of their meaninglessness. In our experiments, we use the Vietnamese stopword dictionary [9] for removing stop words in the sentence.

3.1.3 Machine Learning Model:

Follow An et al study [5], we use some Machine Learning models such as Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine combined with TF-IDF.

Logistic Regression and Naive Bayes Classification are both classification algorithms used to assign objects to a discrete set of values. In which, Logistic Regression uses the sigmoid function to make a probability assessment, Naive Bayes Classification predicts based on probability calculation applying Bayes theorem.

Decision Tree is the most powerful and popular tool for classification, and prediction. A Decision tree is a flowchart like a tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) stores a class label. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from preliminary data(training data).

Random Forest is a machine learning algorithm built on multiple sets of Decision Tree. The model's output is based on the aggregate decision on the decision trees it generates with the voting method. Random Forest is a Supervised Learning method to handle classification and regression problems.

Support Vector Machines (SVM) is a supervised learning method used for classification, regression, and outlier detection. SVM is used to construct a hyperplane in multidimensional space to separate different layers. The goal of SVM is to find the hyperplane that best divides the two classes. The distance between the hyperplanes and the closest data points from each layer is defined as the boundary.

TF-IDF (Term Frequency-Inverse Document Frequency) is a well-known method used to evaluate the importance of a word in a document used for information retrieval and natural language processing, in which the TF component represents the frequency of occurrence of a word in a document, the IDF component represents the inverse of the frequency of the document, used to evaluate the importance of a word in the text. The goal of TF-IDF is to calculate word frequency in text in a huge document pool.

3.1.4 Deep Learning Model

The transfer learning model has attracted increasing attention from NLP researchers around the world because of its outstanding achievements. In this study, we use several models:

BERT4News [10] [11]: This is a language modeling language trained on a dataset consisting of news articles. It has proven to be effective in hate speech detection in news articles.

mBERT Cased [12] [10]: This is a language model trained on a data file consisting of text and code. It has proven to be effective in hate speech detection in code snippets.

mBERT Uncased [12] [10]: This is a large-scale language model like mBERT Cased, but case-insensitive. This means it can handle upper and lower case text.
XLM-R BASE [12] [10]: This is a large language model trained on a data file consisting of texts from many different languages. It has been proven to be effective in hate speech detection in different languages.
XLM-R LARGE [12] [10]: This is a large configurable modeling language like XLM-R BASE, but larger. This means it can learn more information from its training dataset. Therefore, it can be more effective in hate speech detection.
Bi-GRU-LSTM-CNN is a distributed neural network model used for hate speech detection published by Tin et al [1]. This model combines the priorities of bidirectional regenerative neural network (RNN) (Bi-GRU) and convolutional neural network (CNN). Bi-GRU is an RNN that can process text from both sides of a word, which means that Bi-GRU can understand the relationship between words in a sentence. CNN is a neural network that can detect local feature sets in a text and can detect important words or phrases in a sentence.
PhoBERT-CNN is a large language model published by Khanh et al [13]. This model is a combination of PhoBERT based pre-trained model from HuggingFace with the Text-CNN model. The output of PhoBERT based pre-trained is used as input embedding for Text-CNN. The output of PhoBERT based pre-trained is used as input embedding for Text-CNN. PhoBERT based pre-trained is initialized with a max sequence length is 20. Text-CNN is built with four layers of conv1D with filter size is 32 and size 1, 2, 3, 5, respectively. This combined model has an Adam optimizer, the learning rate is 2e-5, epsilon is 1e-8, and dropout is 0.4. This combined model has an Adam optimizer, the learning rate is 2e-5, epsilon is 1e-8, and dropout is 0.4.

3.2 Online Prediction

Online prediction forms an integral part of the system's pipeline, including distinct steps such as data crawling, data streaming via Kafka, and utilizing the best pre-trained model to make predictions.

3.2.1 Crawling Data In this stage, we utilized the Python library "pytchat" to gather data from YouTube. Specifically, we acquired the ID of any random live stream video and initiated a loop within the system. This loop continuously retrieved real-time chat messages (live chat) from the specified video and collected the comments. Subsequently, every set of ten comments was saved as a CSV file and stored in the designated directory.

3.2.2 Preprocessing Similar to the offline learning phase, the collected data will undergo preprocessing to eliminate unnecessary noise characters, enhancing the model's predictive capability.

3.2.3 Kafka Stream Data Apache Kafka is a distributed streaming platform that encompasses essential components such as topics, producers, consumers, brokers,... These components enable Kafka to effectively process data streams, create real-time data pipelines, and handle large-scale data integration seamlessly. In our study, we utilized Kafka as a big data technology to facilitate faster data loading compared to traditional methods.

In more detail, firstly, we initiate a topic named "stream_real_time" on the local server to facilitate data streaming, with the topic being partitioned into 3 partitions to expedite the streaming process.

Table 2. Configuration of Kafka topic

Topic	Partition	Replication Factor	Server
stream_real_time	3	1	127.0.0.1:9092

Next, we write the preprocessed data into the newly created Kafka topic. Using the TensorFlow I/O package, we create a Kafka group I/O dataset, enabling us to stream data from the "stream_real_time" topic in Kafka to the model for making predictions.

3.2.4 Model Prediction We utilize the best-performing model obtained during the offline learning phase to classify hate speech in real-time, with two labels: 'CLEAN' (no hate speech) and 'HATE' (containing hate speech elements). The reason for using the best model from offline learning is to ensure the highest possible accuracy in the prediction results.

4 Result

4.1 Offline Learning Result

4.1.1 Metrics We have used Accuracy and F1-score measures to evaluate the classification performance of the models, where TP is True Positive, TN is True Negative, FP is False Positive, FN is False Negative given in the expressions below:

Accuracy:

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

F1-Score:

$$F1 - Score = \frac{2 * precision * recall}{precision + recall},$$

where

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

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

4.1.2 Experimental Settings We deploy Machine Learning models with the parameters of each model: Naive Bayes with "alpha" = 1.0; Logistic Regression with maxIter = 20, and regParam = 0.3; Decision Tree with max_depth = 17, and min_samples_leaf = 3; Random Forest with numTrees = 200, maxDepth = 10, maxBins = 64; SVM with kernel='rbf', C=1.0, gamma='scale'. The parameters of the Deep Learning models are listed in Table 3.

Table 3. Parameters of Deep Learning models.

Model	Batch Size	Embed Size	Num Epochs	Learning Rate
Bi-Gru-LSTM-CNN	128	300	30	1e-3
PhoBERT-CNN	128	768	20	2e-5
BERT4News	16	768	2	1e-4
mBERT Cased	64	768	5	1e-4
mBERT Uncased	64	768	5	1e-4
XLM-R BASE	64	768	5	5e-5
XLM-R LARGE	16	1024	5	5e-6

4.1.3 Experimental Results In general, the classification results of the offline models give positive results, in which the XLM-R LARGE model gives the best results on both data sets. A clear difference can be seen in the classification performance of Machine Learning and Deep Learning models, the best model according to Deep Learning architecture gives higher results than the best model of Machine Learning. This result corresponds to studies in recent years that have shown that Deep Learning and especially models using BERT architecture often give good results more so with Machine Learning models in most NLP tasks.

Besides, we can see that the Accuracy and F1-Score measures on the HSD-VLSP dataset have a large difference, however, the Accuracy and F1-Score measures on the HSD-VLSP dataset do not have such a big difference. Because the difference between the label ratio on the HSD-VLSP set is quite large, and the label ratio on the ViHSD set is quite similar, it has caused a difference in the two measures when evaluated on the HSD-VLSP dataset.

From the classification results of the models in Table 4, we choose XLM-R LARGE, the model with the best results, for the Online Prediction task on real-time data.

Table 4. Results of training the models.

Models	ViHSD		HSD-VLSP	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**
Naive Bayes	77.9	77.8	93.28	64.38
Logistic Regression	81.2	81.18	94.37	64.69
Decision Tree	76.5	76.5	93.33	64.69
SVM	80.0	79.98	94.15	60.38
Random Forest	81.15	81.5	94.26	62.21
Bi-Gru-LSTM-CNN	79.53	79.53	91.97	66.79
PhoBERT-CNN	81.38	81.38	93.72	67.93
BERT4News	80.14	80.2	94.81	68.55
mBERT Cased	79.58	79.57	94.76	69.38
mBERT Uncase	78.76	78.76	93.88	67.62
XLM-R BASE	83.28	83.26	94.16	67.53
XLM-R LARGE	**84.27**	**84.27**	**94.87**	**70.36**

4.2 Online Prediction Result

The XLM-R LARGE model made predictions on approximately 1000 data points, which are comments from a live stream video on a Vietnamese YouTube channel named "500BROS CS:GO" on July 20, 2023. The predictions were carefully examined and labeled for evaluation. The resulting label distribution indicated a nearly balanced proportion between clean and hate speech labels is shown in Table 5:

Analyzing the model's performance based on the labeled data, we observed

Table 5. The number of actual labels during the online prediction phase

Clean	Hate	Total
512	492	1004

a promising outcome in hate speech detection, as evidenced by the confusion matrix is shown in Table 6:

Table 6. Confusion matrix of online prediction phase

	Clean	Hate
Clean	460	52
Hate	60	432

The accuracy of the model real time prediction is 74.6%, showcasing a substantial portion of correct predictions out of the total. This suggests that the model demonstrated a proficient ability to classify instances with notable accuracy. However, it's worth noting that the model still has some weaknesses, particularly in false positive and false negative cases, which were non-negligible in number compared to the total predictions. Such errors could have significant consequences, either falsely flagging harmless content or overlooking instances of actual hate speech. Therefore, further improvements are crucial to enhance the model's performance and ensure more reliable hate speech detection, fostering a safer online environment. The implications of these results need deeper investigation, as we aim to refine the model to achieve higher accuracy and minimize misclassifications. Subsequent sections will delve into a comprehensive analysis of the model's performance, offering valuable insights for future research and development in the domain of hate speech detection.

5 Discussion

5.1 Error Analysis

During real-time prediction, we identified a significant underlying issue. This issue stems from the presence of certain words that, when written without diacritics, can be interpreted ambiguously, leading the model to misunderstand and misclassify them as hate speech. For example, the phrase "choi lon," which means "play big," could be misinterpreted as "choi l*n," a vulgar term. This ambiguity poses a challenge to the model's accurate classification. Furthermore, some comments, although devoid of explicit offensive words, exhibit clear hate speech tendencies. These instances present another layer of complexity for the model's performance as it struggles to recognize implicit hate speech.

Table 7. Several examples of classification error on the given datasets

Comment	**Actual Label**	**Predict Label**
Choi 1 cu lon thoi nao cac giao su (English: Let's play big once)	CLEAN	HATE
Sút hụt 3 trái liên tiep là hay du chua (English: Missing three consecutive shots, is it skillful or aggressive?)	HATE	CLEAN
Choi thê thì chịu luôn roi (English: I'm sick of how you are playing)	HATE	CLEAN
Lam sao de dm den 500BROS v ae (English: Hey guys, how to send direct message to 500BROS?)	CLEAN	HATE

5.2 System Latency

In real-time prediction systems, minimizing latency is of paramount importance and is always a top priority. Within our system, two stages have the potential to cause latency, which includes "data loading speed" and "prediction speed".

5.2.1 Data Loading Speed

In the real-time data collection process, we utilized two attributes: "chat.datetime" - storing the timestamp of each comment as provided by the pytchat library, and "datetime.now()" - representing the current timestamp offered by the datetime library. These attributes enabled us to compare data collection speed effectively. As a result, our observations revealed that the time taken from the moment a comment appeared in the live chat video on YouTube to the point of its collection was approximately 1.76 seconds. Consequently, it is evident that the data collection speed was exceptionally swift.

5.2.2 Prediction Speed

For the model prediction phase, we also utilized the "datetime.now" attribute at two different time points: first, after streaming the data into the model, and second, when the model generated the predictions. Our results indicated that, on average, each comment took approximately 2.1 seconds for the model to make predictions. Please note that the latency mentioned was not caused by the Kafka streaming pipeline, it originated from the model's prediction process. While some comments were predicted very quickly, there were instances where the model took longer, sometimes nearly 4 seconds. This discrepancy can be attributed to several factors. For instance, some comments were very short and contained only a few offensive words, making it easy for the model to predict their classification. On the other hand, longer comments required more time for the model to process, or they might fall into cases that the model hadn't been trained on, leading to longer prediction times

6 Conclusion and Future Work

6.1 Conclusion

In this study, we presented a real-time hate speech prediction system of online data on Youtube platform. The recommendation system is developed using Kafka and Machine Learning, Deep Learning models. The system consists of two components, an offline hate speech analysis model and an online prediction pipeline.

In the offline component, we train and evaluate the model on two Vietnamese hate speech detection datasets including comments on social networks, then find the best model, which will be used to predict hate speech for the comments collected in real time. We have successfully installed Machine Learning and Deep Learning models and evaluated them on two metrics, Accuracy and F1-Score, in which the XLM-R Large model with a more complex, modern model architecture gave the best results with...

The other component is an online prediction pipeline used to predict the hate speech of comments in real time collected from live streaming videos on the Youtube platform. Specifically, this pipeline uses Kafka to collect real-time comments, then this data will be preprocessed and applied the pre-trained best model, XLM-R Large, to predict the label as HATE or CLEAN. In summary, we have successfully built a real-time data processing system and applied Deep Learning model on Apache Kafka big data processing engine.

6.2 Future Work

In the future, we hope to improve and test more models for better classification performance. In addition, we also plan to expand the scope of our research, doing this work on other Vietnamese language datasets with many different topics and expanding the online data source because besides Youtube, there are many social networks or online data sources that are also very popular. In addition, we also want to be able to collect and process real-time data on multiple online videos at the same time. Therefore we also hope to be able to solve these problems to increase the performance and speed of the system we have built.

References

1. T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate speech detection on vietnamese social media text using the bi-gru-lstm-cnn model," *arXiv preprint arXiv:1911.03644*, 2019.
2. H. T.-T. Do, H. D. Huynh, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate speech detection on vietnamese social media text using the bidirectional-lstm model," *arXiv preprint arXiv:1911.03648*, 2019.
3. S. N. T. L. J. Xuan-Son Vu, Thanh Vu, "Etnlp: A visual-aided systematic approach to select pre-trained embeddings for a downstream task," *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, 2019.
4. H. H.-P. Vo, H. H. Nguyen, and T.-H. Do, "Online hate speech detection on vietnamese social media texts in streaming data," *Artificial Intelligence in Data and Big Data Processing: Proceedings of ICABDE 2021*, vol. 124, p. 315, 2022.
5. L.-A. Doan, P.-T. Nguyen, T.-O. Phan, and T.-H. Do, "An implementation of large scale hate speech detection system for streaming social media data," in *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT).* IEEE, 2022, pp. 155–159.
6. S. T. Luu, K. V. Nguyen, and N. L.-T. Nguyen, "A large-scale dataset for hate speech detection on vietnamese social media texts," in *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, July 26–29, 2021, Proceedings, Part I 34*. Springer, 2021, pp. 415–426.
7. X.-S. Vu, T. Vu, M.-V. Tran, T. Le-Cong, and H. Nguyen, "Hsd shared task in vlsp campaign 2019: Hate speech detection for social good," *arXiv preprint arXiv:2007.06493*, 2020.
8. T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "Vncorenlp: A vietnamese natural language processing toolkit," *arXiv preprint arXiv:1801.01331*, 2018.
9. (2017) Van-duyet le. [Online]. Available: https://github.com/stopwords/vietnamese-stopwords
10. S. T. Luu, K. Van Nguyen, and N. L.-T. Nguyen, "Impacts of transformer-based language models and imbalanced data for hate speech detection on vietnamese social media texts," 2022.

11. K. N. T. Thanh and K. N. Van, “ReINTEL challenge 2020: Exploiting transfer learning models for reliable intelligence identification on Vietnamese social network sites,” in *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*. Hanoi, Vietnam: Association for Computational Lingustics, Dec. 2020, pp. 45–48. [Online]. Available: https://aclanthology.org/2020.vlsp-1.9
12. S. N. Hoang, B. Nguyen, N. P. Nguyen, S. T. Luu, H. T. Phan, and H. D. Nguyen, “Enhanced task-based knowledge for lexicon-based approach in vietnamese hate speech detection,” in *2022 14th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2022, pp. 1–5.
13. K. Quoc Tran, A. Trong Nguyen, P. G. Hoang, C. D. Luu, T.-H. Do, and K. Van Nguyen, “Vietnamese hate and offensive detection using phobert-cnn and social media streaming data,” *Neural Computing and Applications*, vol. 35, no. 1, pp. 573–594, 2023.