# ReINTEL Challenge 2020: A Comparative Study of Hybrid Deep Neural Network for Reliable Intelligence Identification on Vietnamese SNSs

**Hoang Viet Trinh, Tung Tien Bui, Tam Minh Nguyen**
**Huy Quang Dao, Quang Huu Pham, Ngoc N. Tran**
trinh.viet.hoang@sun-asterisk.com
AI Research Team, R&D Lab, Sun* Inc.

**Ta Minh Thanh**
Le Quy Don Technical University, Ha Noi, Vietnam

## Abstract

The overwhelming abundance of data has created a misinformation crisis. Unverified sensationalism that is designed to grab the readers' short attention span, when crafted with malice, has caused irreparable damage to our society's structure. As a result, determining the reliability of an article has become a crucial task. After various ablation studies, we propose a multi-input model that can effectively leverage both tabular metadata and post content for the task. Applying state-of-the-art fine-tuning techniques for the pretrained component and training strategies for our complete model, we have achieved a 0.9462 ROC-score on the VLSP private test set.

## 1 Introduction

### 1.1 Overview

The fast growth of social media and misinformed contents have posed an incremental challenge of exposing untrustworthy news to billions of their global users, including 65 million Vietnamese users (Social, 2020). Consequently, the spread of mistrust information on social cites has placed real damages on government, policymakers, organizations, and citizens of many countries (Cheng and Chen, 2020; Pham et al., 2020), resulting in an urge for fast and large-scale fact-checking online contents. With the enormous amount of news and information on the internet daily, this is impossible to be efficiently done only by human efforts, putting a quest to create a trustworthy system to perform the task automatically.

Reliable Intelligence Identification on Vietnamese SNSs (ReINTEL) is the task of reliable or unreliable social-network-sites (SNSs) identification. The main difficulties of these tasks, including:

- The given data (contents of social sites) is unstructured, containing mostly texts combined with metadata (including: images, dates, numbers, username, id, *etc*). The meta-information is partially missing and incorrect, making the usage of those data more challenging.
- The problem is multi-modal learning, which 'involves relating information from multiple sources' (Sachowski, 2016), resulting in the search for a proper combination of features from those sources to learn a unified model with high performance.

### 1.2 Our contributions

In this paper, we propose our methods to resolve these above-mentioned problems. With thorough experiments, we determined to answers two main questions: Should we incorporate multi-source data? Furthermore, how to combine them in terms of training strategies? Our contributions are as followed:

- We provide a reliable method of data cleansing, making metadata ready for prediction.
- More importantly, we are the first who construct a comprehensive comparative study to discover the effectiveness of models when incorporating multi-source data with different training strategies. Our experiment's results reveal that:
  - Models using text or meta-features alone has a crucial gap in performance, indicating that texture information is significantly more predictive than metadata.
  - Models utilize multi-source data with different training strategies results in a wide range of performance. This finding implies that combining data in training has a significant impact on the overall performance.
  - Combining data from multi-sources with particular training plans leads to our best

models. Additionally, the model trained with metadata alone performs significantly better than a random guess, shedding light on the meta data's informativeness.

- We apply state-of-the-art transfer learning methods for textual feature extractions and neural network (in comparison with other traditional machine learning methods) for tabular-data feature representation, achieving the competitive performance of 0.9418 ROC-score on the public test set (ranked 2nd) and 0.9462 ROC-score (ranked 3th) on the private test set.

### 1.3 Roadmap

In the following sections, we briefly review some related works involve with our methods. Next, in section 3, we illustrate our method in detail. Our experiments are described in Section 4, including dataset description, data preprocessing methods, and our model configurations, whereas Section 5 indicates all of our experimental results. Finally, section 6 is the conclusion for our proposed framework.

## 2 Related work

### 2.1 Contextual Representation For Text

Recent works on learning universal representation for text, namely Elmo (Peters et al., 2018), GPT (Radford, 2018), BERT (Devlin et al., 2018) have brought remarkable improvements for wide, diverse NLP downstream tasks: Text Classification, Question Answering and Named Entity Recognition. In contrast to traditional methods such as Word2vec (Mikolov et al., 2013) or Glove (Pennington et al., 2014) which learns context-independent word embeddings, universal language models were trained on a massively large amount of unlabeled data with different pretext tasks, including causal language modeling and masked language modeling, to learn a deep contextual representation of words given its context.

### 2.2 Fake News Detection on SNSs

Studies of fake news identification on social network sites have gained significant attention recently. Most of them utilize data from multiple sources. For example, CSI (Ruchansky et al., 2017), a framework with several modules based on Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and a fully connected layer that utilizes the article's contents, the users' responses and behaviors of source users who promote it. Another instance is dEFEND (Shu et al., 2019), which exploits both news contents and user comments with a deep hierarchical co-attention network to learn a rich representation for fake news detection. From a slightly different point of view, TriFN (Shu et al., 2017) models a tri-relationship between users, publishers, and new contents by several embedding methods and experiments promising results.

Table 1: Statistics of the datasets.

| | Dataset |
|---|---|
| Total News | 5172 |
| Users | 3706 |
| Unique News | 5087 |
| News have images | 1287 |
| Reliable News | 4238 |
| Unreliable News | 934 |

Although utilizing multi-source data, existing research appears to lack a comprehensive study on the effectiveness of input-combination strategies.

### 2.3 Vietnamese Natural Language Processing

Inspired by BERT's textual learning methods, PhoBERT (Nguyen and Nguyen, 2020) was proposed to extend the successes of deep pre-trained language models to Vietnamese. Its pretraining approach is based on RoBERTa (Liu et al., 2019) training strategies to optimize BERT training procedure. Additionally, PhoBERT also consists of two different settings, PhoBERT Base, which uses 12 Transformer Encoder layers and 24 layers with PhoBERT Large. It improves many Vietnamese NLP downstream tasks. For instance, Pham (Pham et al., 2020) introduced novel techniques to adapt general-purpose PhoBERT to a specific text classification task and archives state of the art on Vietnamese Hate Speech Detection (HSD) campaign.

## 3 Methodology

### 3.1 Dataset

In this paper, we use the dataset provided by VLSP organizers for ReINTEL task (Le et al., 2020), composed of contents from Vietnamese social network sites (SNSs), e.g., Facebook, Zalo, or Lotus (Social, 2020). There are approximately

5,000 labeled training examples, while the test set consists of 2,000 unlabeled examples. Each example is provided with information about the news's textual content, timestamp, number of likes, shares, comments, and attached pictures. Table 1 indicates the detailed statistic of the dataset, the data distribution of reliable and unreliable news was heavily imbalanced and skewed toward trustworthy contents.

### 3.2 Data preprocessing

Fake news can be studied with respect to four perspectives: (i) knowledge-based (focusing on the false knowledge in fake news); (ii) style-based (concerned with how fake news is written); (iii) propagation-based (focused on how fake news spreads); and (iii) credibility-based (investigating the credibility of its creators and spreaders) (Zhou and Zafarani, 2018). In this task, with the ReINTEL dataset, we focused on knowledge-based and credibility-based. Specifically, we performed the following preprocessing to extract the necessary information.

- **Deleted incorrect data rows**: While mining data, there are few incorrect rows due to the process of collecting and storing data. We decided to delete these rows from the data set.

- **Filled missing value**: To deal with missing values, we fill them with different strategies: numbers with 0, timestamps with the min timestamp and post messages with empty string

- **Extracted date time features from timestamp values**: For each timestamp value, we decoded these to date time values to enrich feature: minutes, hours, days, months, years, weekdays, etc.

- **Created `user_score` feature**: For user id, we created a user reputation score metric based on previous posts in dataset. This score is used to evaluate the user's future posts

- **Created `image_count` feature**: With images of each post, we compiled several information, including: number of images and image's aspect ratio

- **Preprocessed `post_message` feature**: We perform post messages preprocessing more carefully than the rest. The processing stages are listed below:
  - Filled missing value with empty string
  - Standardized Vietnamese punctuation
  - Removed HTML tags
  - Replaced email, links, phone, numbers, emoji, date time with new corresponding token

### 3.3 Model for Tabular Data

Metadata for the ReINTEL dataset is composed of all input features except post message (text data). We tried numerous machine learning algorithms to learn a classifier using only metadata, ranging from traditional methods: Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbor, Decision Tree, Gaussian Naive Bayes, Support Vector Machine, Adaptive Boosting, Gradient Boosting, Random Forest (Hastie et al., 2001), and Extra Trees (Geurts et al., 2006) to a deep learning method: Multi-Layer Perceptron (Hastie et al., 2001)

We then proceeded to select a handful of model with high performances and complexities to serve as a base model for stacking (Wolpert, 1992). Meanwhile, for the meta-model used in stacking, we chose Logistic Regression. We also did the same for blending ensemble (Sill et al., 2009).

### 3.4 Deep learning-based Content Classification

BERT's layers capture a rich hierarchy of linguistic information, with surface features at the bottom, general syntactic knowledge in the middle, and specific semantic information at the top layer (Jawahar et al., 2019). Therefore, in order to better benefit for our downstream task, we incorporate as much as possible different kinds of information from our model backbone PhoBERT by concatenating [CLS] hidden states from each of 12 blocks, followed by a straightforward custom head, which is a multilayer perceptron with Dropout (Srivastava et al., 2014). The architecture of the model is shown in the Figure 1.

### 3.5 Deep Multi-input Model

Our experiments (details are in the below section) indicates that meta data is informative predictors for reliable and unreliable news classification. Therefore, we decided to combine both text and meta data to resolve the task. The structure of our

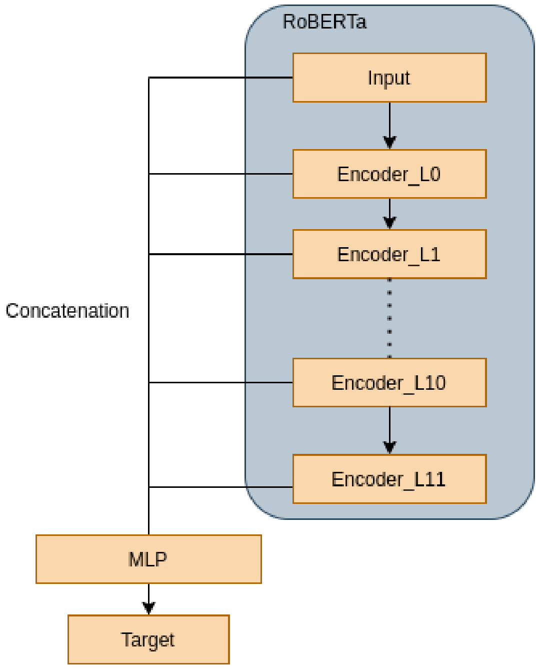

Figure 1: The architecture model for content classification using RoBERTa pre-trained model.

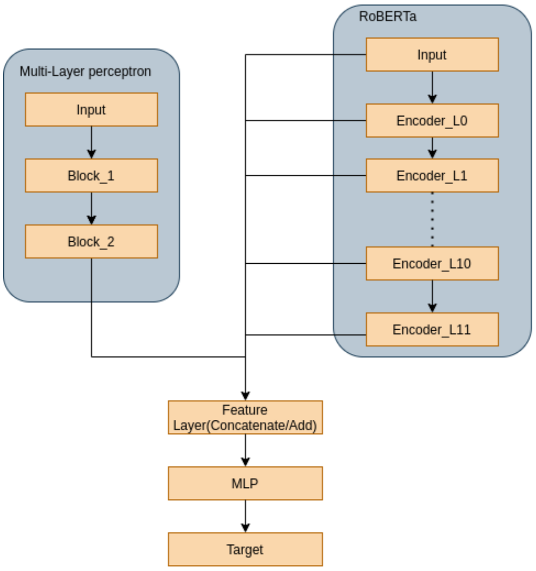

Figure 2: An illustration of our proposed deep multi-input architecture.

multi-input model is described (in Figure 2) as followed: output features of Multi-Layer Perceptron and RoBERTa models, after being concatenated or added together, were simply passed through a custom head classifier.

# 4 Experiments

## 4.1 Model Settings

We divide the dataset into a training set and a validation set with 10-fold cross validation method. Each fold, we use AdamW (Kingma and Ba, 2014) for optimization with a learning rate of $10^{-5}$ and a batch size of 32. Warm-up learning was applied, with the chosen maximum learning rate was $2 \times 10^{-5}$. Except for all bias parameters and coefficients of LayerNorm layers (Ba et al., 2016), the rest of the model's parameters were regularized with weight decay to reduce overfitting. We used a regularization coefficient of 0.01. The number of training epochs was 20.

Instead of using cross-entropy loss, we implemented a label smoothing cross-entropy loss function, a combination of cross-entropy loss and label smoothing (Müller et al., 2019). The smoothing rate is set to 0.15.

## 4.2 Fine-tuning technique

We applied state-of-the-art fine-tuning techniques including: gradual unfreezing, discriminate learning rate, warm-up learning rate schedule (Pham et al., 2020) to perform effective task adaptation (Gururangan et al., 2020).

## 4.3 Training Strategies

We apply four training strategies to study the effects of combining text and mate data on our above-mentioned multi-data model's performance. Notice here that we used the pre-trained weights of RoBERTa as the initialization for the textual-feature-extraction-model's backbone in all strategies. We refer to the textual and meta feature extraction parts of the multi-source model are referred as text and meta submodel for short. Our training policies are described as followed:

- Strategy 1 (S1): The parameters of both the text submodel's head and the meta submodel are initialized randomly

- Strategy 2 (S2): The meta submodel will be trained for the task first. Its feature extraction part (all layers except the output one used for classification) is used to combine with the text submodel. The parameters of the text submodel's head are initialized randomly.

- Strategy 3 (S3): Meta submodel is un-trained when incorporates with the text submodel, which is already fine-tuned with the task.
- Strategy 4 (S4): Both the two submodels are trained/fine-tuned with the classification task before being combined for further training.

### 4.4 System configuration

Our experiments are conducted on a computer with Intel Core i7 9700K Turbo 4.9GHz, 32GB of RAM, GPU GeForce GTX 2080Ti, and 1TB SSD hard disk.

## 5 Experimental Results

### 5.1 Evaluation metrics

For this work, we used the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), a common evaluation metrics for classification tasks. The Receiver Operating Characteristic (ROC) curve shows how well a model classify samples by plotting the true positive rate against the false positive rate at various thresholds. To turn the graph into a numerical metrics, the Area Under Curve (AUC) is then evaluated. A maximum value of 1.0 indicates that the model predicts correctly for all thresholds, and a minimum of 0.0 implies the model gets everything wrong all the time. The formula for ROC-AUC is

$$\text{ROC-AUC} = \int_{0}^{+\infty} \int_{-\infty}^{+\infty} f_1(u) f_0(u-v) du dv \tag{1}$$

where $f_1$ and $f_0$ are the density functions.

### 5.2 Our results

Our results are shown in Table 2 3 4 5

Table 2 compares the effectiveness of traditional machine learning algorithm on metadata. The performance ranges from a ROC-AUC score of 0.5450 with a simple Logistic Regression, to 0.7338 through employing Gradient Boosting across various models. Despite achieving results not as competitive as which of Gradient Boosting, the Multi-Layer Perceptron model was chosen due to its differentiability, which enabled joint training with the textual model (details in Section 3.5).

Table 2: Performance of models using only meta data.

| Method | ROC-AUC |
|---|---|
| Logistic Regression | 0.545037 |
| Linear Discriminant Analysis | 0.545037 |
| K Nearest Neighbors | 0.633251 |
| Decision Tree | 0.657217 |
| Gaussian Naive Bayes | 0.588978 |
| Support Vector Machine | 0.599256 |
| Adaptive Boosting | 0.673511 |
| Gradient Boosting | **0.733850** |
| Random Forest | 0.727192 |
| Extra Tree | 0.651323 |
| Multi-Layer Perceptron | 0.604653 |

Table 3: ROC-AUC score on public test of combining feature from blocks. Input model is the text content of the news.

| Blocks | ROC-AUC |
|---|---|
| Block 1-6 | 0.913251 |
| Block 6-12 | 0.937330 |
| Block 9-12 | 0.921147 |
| Block 1-12 | 0.939915 |
| Block 1-12 (Ensemble) | **0.941811** |

Most of the aforementioned model's performances are significantly better random guessing, indicating that metadata is an informative predictor for the news classification task.

Table 3 shows the ROC-AUC scores as we tried incorporating different embeddings from different RoBERTa blocks. Specifically, as illustrated in Figure 1, we selected a subset of all embeddings RoBERTa generated, which are then concatenated together and passed through a classifier. Amongst our trials, an ensemble of various combinations across all embeddings achieved the highest AUC-ROC score of 0.9418.

Table 4 highlights one of the major discoveries of our work. It presents our best results for models using only meta- or text data to classify SNS. The

Table 4: Performance of models using only either text or meta data.

| Blocks | ROC-AUC |
|---|---|
| Only meta data | 0.7338 |
| Only text data | **0.9628** |

Table 5: Performances of multi-data model with different training strategies.

| Blocks | ROC-AUC |
|---|---|
| Strategy 1 (S1) | 0.9058 |
| Strategy 2 (S2) | 0.9399 |
| Strategy 3 (S3) | 0.9552 |
| Strategy 4 (S4) | **0.9628** |

performance gap between the two models is significant (more than 0.20 in ROC-AUC score), pointing out that textual features are more predictive than metadata. Besides, using only meta-features is considerably more accurate than random guess (0.7338 ROC-AUC score), indicating that its information can be employed to train a better model.

Table 5 sheds lights on how to effectively combined multi-source data. S1, S2, S3, and S4 in the table refer to the previously-mentioned strategy 1, strategy 2, strategy 3, and strategy 4. S1 and S2 result in the least performance among the four, less than almost 0.05 and 0.02 ROC-AUC score than our second best strategies, S4. Additionally, compared to training with only textual features even better than S1 and inconsiderably worse than S2. This result indicates that fine-tuning text submodel with the task before combining with meta submodel is crucial to achieving high performance.

The worsen results of S1 compared to S2 and S3 compared to S4 points out that pretraining meta submodel before the combination of 2 submodels enhances the overall training.

## 6 Conclusion

This paper has constructed a comprehensive comparative study to discover the effectiveness of models with multiple inputs and mixed data. We have explored and proposed different training strategies to train the hybrid deep neural architecture for reliable intelligence identification task. By conducting experiments using PhoBERT, we have demonstrated that combining mixed data with particular training plans leads to our best results. With our proposed methods, we have achieved a competitive performance of 94.18% ROC-score on the public test and 94.62% ROC-score on the private test set in VLSP's ReINTEL 2020 campaign.

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Yang Cheng and Zifei Fay Chen. 2020. The influence of presumed fake news influence: Examining public support for corporate corrective response, media literacy interventions, and governmental regulation. *Mass Communication and Society*, 23(5):705–729.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Duc-Trong Le, Xuan-Son Vu, Nhu-Dung To, Huu-Quang Nguyen, Thuy-Trinh Nguyen, Linh Le, Anh-Tuan Nguyen, Minh-Duc Hoang, Nghia Le, Huyen Nguyen, and Hoang D. Nguyen. 2020. Reintel: A multimodal data challenge for responsible information identification on social network sites.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. When does label smoothing help?

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Quang Pham, Nguyen Viet Anh, Linh Doan, Ngoc Tran, and Ta Thanh. 2020. From universal language model to downstream task: Improving roberta-based vietnamese hate speech detection.

A. Radford. 2018. Improving language understanding by generative pre-training.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news. *CoRR*, abs/1703.06959.

Jason Sachowski. 2016. *Identify Potential Data Sources*, pages 63–72.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. Defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, page 395–405, New York, NY, USA. Association for Computing Machinery.

Kai Shu, Suhang Wang, and Huan Liu. 2017. Exploiting tri-relationship for fake news detection. *CoRR*, abs/1712.07709.

J. Sill, G. Takács, L. Mackey, and D. Lin. 2009. Feature-weighted linear stacking. *ArXiv*, abs/0911.0460.

We Are Social. 2020. Digital 2020 - global digital overview.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241 – 259.

Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *CoRR*, abs/1812.00315.