Explainable Deep Learning for Thai Stock Market Prediction Using Textual Representation and Technical Indicators

Tanawat Chiewhawan
Chulalongkorn University Big Data Analytics and IoT Center (CUBIC)
Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University
6170929321@student.chula.ac.th

Peerapon Vateekul
Chulalongkorn University Big Data Analytics and IoT Center (CUBIC)
Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University
peerapon.v@chula.ac.th

ABSTRACT
In this paper, we proposed a deep neural network to predict the Thailand stock market (SET index) with the capability to analyze both numerical and textual inputs altogether. The datasets include twelve-years market data with generated technical indicators (73 time-series) and Thai economic news headlines from various online sources. The proposed model consists of a long-short term memory model and textual representations to predict one-day-ahead percentage market changes. We experiment on two textual representation approaches, which is the Hierarchical neural network and Bidirectional Encoder Representations from Transformers (BERT) together with aggregated embedding. The three-year average experimental results show that adding textual representation increase profit of 6.9% on the proposed strategy-free metric. Next, the Hierarchical approach performs considerably better than the BERT method showing up to 4.4% profit increase. Finally, we demonstrate another advantage of the hierarchical approach for model interpretability using the integrated gradient attribution methods. This interpretability lets us analyze the underlying textual relevance and also for future model improvement.

CCS Concepts
•Computing methodologies→Neural networks • Computing methodologies→Natural language processing.

Keywords
Deep learning; natural language processing; stock prediction.

1. INTRODUCTION
The stock market prediction task is challenging due to its volatility and varieties of factors that affected them. Recently, deep learning technology has gained more popularity in many stocks prediction research. Most of them focused on either numerical data (such as price information, fundamental values, technical indicators) or textual data (such as financial news and reports). However, investors can analyze the market with both types of information before making any decision. Correspondingly, in Thailand stock market data, there are only a few deep learning research, and most of them [1; 2] focused on sentiments classification or utilized either numeric or textual features. To the best of our knowledge, we are the first to extend the study for the Thai stock market combining the Natural Language Processing (NLP) domain with numeric time-series (technical indicators). Apart from Thailand market data, there are many pieces of research [3-5] already studied those combination for stock prediction showing promising results. Moreover, most of them agreed that using the news headlines or the news titles is sufficient instead of using the whole article.

However, for the Thai language, some of the proposed methods could not be implemented. For example, the usage of Open IE and Event embedding [6] in [4; 5] show promising results in S&P500, but the Open IE is not available for the Thai language. Also, another challenge in Thai NLP is the character and word are written unsegmented (continuously without any delimiters)[7] and could eventually affect model performance. Consequently, in this paper, we examine two alternative textual representation approaches. Firstly, the hierarchical neural network from DeepClue [8] which is a tokenized based model. Secondly, applying one of state of the art in NLP, the BERT [9] of which we utilized the pre-trained multilingual version (a character-based model).

Finally, one of the crucial aspects of using the deep learning model is explainability. Most of the developed models are often viewed as a black box, causing doubt to utilize model prediction fully. We demonstrate the qualitative usage of the Integrated Gradient method [10] over the hierarchical approach to understand the textual relevance of the model over the stock prediction target.

2. RELATED WORKS
Deep Learning with Textual Information. News article had been one of the significant influences on deep learning stock prediction. Shi, et al. [8] proposed a framework for stock return prediction with news input using the hierarchical neural network structure. Their model embedded textual information into three layers of representation, they design this to support model visual interpretability. This hierarchical structure will be one of our textual representation experiments. Recently, BERT [9] is one of the breakthrough states of the art model in NLP, leading to significant advancement in a variety of tasks. BERT architecture consists of Transformers, an attention mechanism built to learns

ICCCM'20, July 17–19, 2020, Singapore, Singapore

ACM ISBN 978-1-4503-8766-8/20/07…$15.00
https://doi.org/10.1145/3411174.3411191

contextual relations between words. In 2019, some researcher start applying the BERT to the financial domain, Araci [11] proposed the finBERT, which is a pre-trained language model on the financial corpus, it significantly improves sentiment accuracy. Likewise, in this paper, we aim to experiment with this state of the art, even though our BERT is not directly a sentiment analysis task. We modified the technique to suit with our stock prediction task, as described in section 3.

Deep Learning with Textual and Numerical Information. Most of the stock prediction researches focus on either textual or numerical information as an input, but not both. Vargas, et al. [4] work is one of the early researches that explores this idea of combining two types of data. The model consisted of a textual input layer with the word2vec embedding and technical indicators (TI) layer. They concatenated these two layer's outputs before feeding to the Long Short-Term Memory (LSTM) for stock trend classification. Their work compared extensively with multiple combinations of baseline, mostly derived from Ding, et al. [6]. Next, Akita, et al. [3] and Oncharoen and Vateekul [5] are other examples of utilizing both types of data. From our observation, most researches show successful performance using the concatenating technique to combine numerical and textual data within the proposed deep learning model.

3. METHODOLOGY

This section will describe our proposed frameworks, as shown in Figure 1. First, we defined the data preprocessing process in section 3.1. Next, in section 3.2 and 3.3, we will discuss our two proposals to handle textual representations, the hierarchical neural network, and the BERT aggregated embedding approach. Finally, section 3.4 illustrates the full model, which combines both type of data.

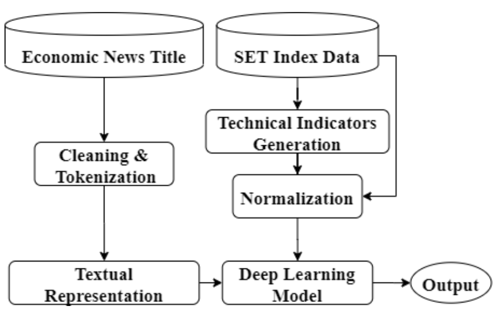

Figure 1. Proposed framework.

3.1 Data Preprocessing

3.1.1 Data labeling

The model prediction target is the next day return ratio of the Stock Exchange of Thailand market index (SET index) as (1).

$$y_t = (c_{t+1} - c_t)/c_t \quad (1)$$

where y_t is the return ratio with features up to time step t,. c_t and c_{t+1} are the close price of time step t and t+1, respectively.

3.1.2 Technical indicators generation

We use Open, High, Low, Close, and Volume (OHLC, V) data to generate technical indicators with the default setting from the "ta" package [12]. As a result, there are total of 73 time-series, including OHLC and V for numeric inputs. Table 1 lists our technical indicators. We normalized all time-series features with the z-score normalization since each input has a different range of value.

Table 1. Technical indicators.

Trend	PSAR	Williams R	Chaikin Money Flow
MACD	Ichimoku	Stochastic Oscillator	Ease Of Movement
ADX	KST	Awesome Oscillator	Force Index
Aroon	**Momentum**	**Volatility**	Negative Volume Index
CCI	KAMA	Average True Range	On Balance Volume
DPO	MFI	Bollinger Bands	Volume Price Trend
EMA	ROC	Donchian Channel	**Other**
Mass Index	RSI	Keltner Channel	Cumulative Return
TRIX	TSI	**Volume**	Daily Log Return
Vortex	Ultimate Oscillator	Acc Dist Index	Daily Return

3.2 Hierarchical Neural Network Approach

The first textual representation approach for this paper is the hierarchical neural network approach. We customed the architecture from the DeepClue paper [8]. We will present only the element that differs from the original work. Foremost, we used the "Newmn" Thai language tokenization from pyThaiNLP framework. Next, we replaced the word2vec embedding with thai2fit embedding [13]. Finally, we change the original prediction layer to a dense layer outputting vector with dimension of [1 x hidden size]. Figure 2 shows the architecture of the hierarchical textual representation approach.

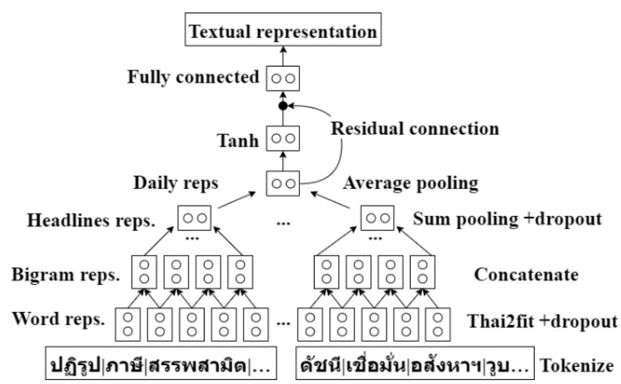

Figure 2. Hierarchical neural network structure [8].

3.3 BERT Embedding Approach

For the second approach, we convert Thai character into numeric vector value using the BERT model. We utilized the pre-train weights of the BERT-Base Multilingual Cased version (mBERT) to fine-tune the language model with our 885K news headlines corpus. After fine-tuning them for three epochs, we input each of the news headlines into the BERT model. We select the first token from mBERT output (pooled hidden vector) [1 x 768], which often used for classification tasks to represent a news headline. Next, we proposed an aggregation method to handle the resource constraints on the GPU memory because there is multiple news occurred per day that needed to be processed (up to 1275 in our datasets). As shown in Figure 3, We aggregate all the news headlines embedding vector published on the same day with either summation, averaging or maximum, results in a single day news headline aggregated representation. This aggregation method might not be perfect because the backpropagation process of deep learning did not update the parameters within the mBERT embedding itself; it treats the aggregated vector as a static input into the deep learning model.

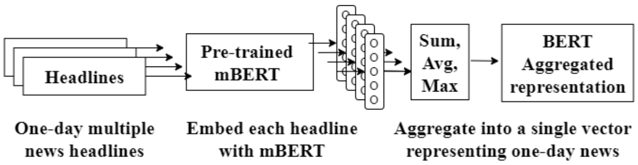

Figure 3. BERT aggregation process for multiple news to represent a single day textual embedding.

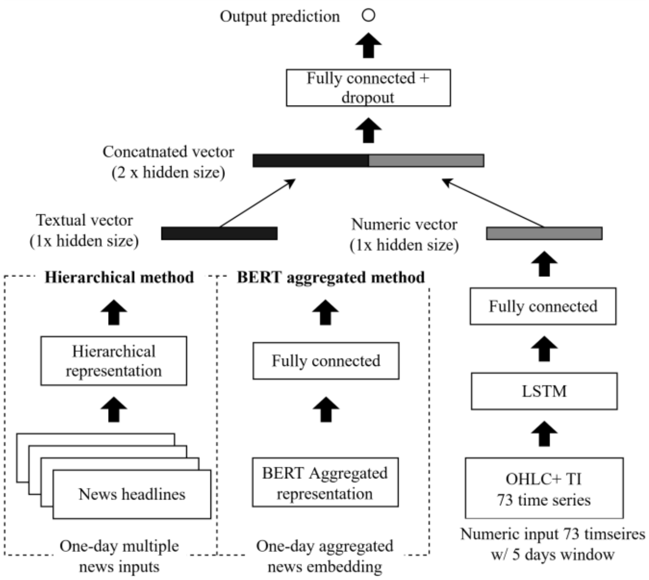

Figure 4. The combination of textual representations and LSTM handling numeric features (price and technical indicators).

3.4 Full Textual and Numeric Model

Following the works like [3-5], we concatenate the textual representation vector with the numeric vector output from the LSTM. Both vectors have a dimension of [1 x hidden size] results in a concatenate vector with dimension of [2 x hidden size] before output the target return ratio from the fully connected prediction layer. Figure 4 illustrates our full proposed model with two options for textual representation approaches.

4. EXPERIMENTAL SETUP

4.1 Dataset

The data in our research consists of two parts, stock market data, and Thai economic topic news headlines. First, we gather the historical data of the SET index corresponding to the period from the 1st of January 2008 to 31st of December 2019. Then, for the same period, we scrapped the news headlines from online sources and filtered for the economic topic only. Next, we split the total of 885K headlines and 2,931 trading days to train/validate/test with three sliding window sets, as summarized in Table 2. We use the daily frequency due to the availability of the news data.

Table 2. Data record summary.

	Data period	Jan-2008 to Dec-2017	Jan-2009 to Dec-2018	Jan-2010 to Dec-2019
News headline records	Training	600,220	631,296	604,764
	Validating	75,425	73,923	72,686
	Testing	73,923	72,686	62,756
Trading days	Training	1,952	1,949	1,950
	Validating	244	244	247
	Testing	244	247	244

4.2 Baseline Model

We proposed comparing methods and baselines listed below.

4.2.1 Random prediction model (RANDOM)

To benchmark the model on randomness, we randomly predict the SET index return each day (1000 simulations) using the historical distribution before the testing period.

4.2.2 Numerical input only (LSTM)

Baselines to evaluate the model performance when using only numerical time-series, the architecture is similar to the full model in Figure 4 but neglects the textual representation parts. Also, the time-series used are OHLC (4 time-series) and OHLC+TI (73 time-series) to evaluate the performance of the model when supplementing with technical indicators.

4.2.3 Textual input only (HIERARCHY or BERT)

In contrast to 4.2.2, these baselines contain only the textual representation neglecting the time-series featured LSTM part. Instead of textual representation vector, HIERARCHY and BERT will output regression value directly.

4.2.4 The proposed method (HIERARCHY or BERT +LSTM TI)

These two are proposed models with combined textual and numeric features inputs, but different in the textual representation approach, the full models shown in Figure 4.

4.3 Evaluation Metrics

We use three metrics to evaluate model including root mean squared error (RMSE), market direction accuracy, and hit profit. The latter two are not common, thus we define them as follows;

4.3.1 Market direction accuracy

This metric evaluates if the predicted return is correct and aligns with the market direction. For example: if tomorrow the actual SET index returns +0.1%, but the model's prediction is negative at -0.2%. We count this as a miss (wrong direction) even the regression error is low.

$$Accuracy\ \% = \frac{\sum_{t=1}^{t=n}(h_t)}{n} \times 100\ \% \quad [14] \qquad (2)$$

$$h_t = \begin{cases} 1 & \text{if } \hat{y}_t \text{ and } y_t \text{ are both positive or negative,} \\ 0 & \text{other wise,} \end{cases}$$

where h_t is the hit count if our predicted direction is correct, n is the number of predictions, y_t is the actual return of the SET index ,and $\hat{y}_t$ is model prediction at the time step t.

Table 3. Model performance summary for three testing-year data splits.

	RMSE (%)				Accuracy (%)				Hit profit (%)			
Model	**2017**	**2018**	**2019**	**Average**	**2017**	**2018**	**2019**	**Average**	**2017**	**2018**	**2019**	**Average**
RANDOM	1.481	1.604	1.522	1.536	50.20	50.00	50.10	50.10	0.20	-0.40	0.30	0.03
LSTM(OHLC)	0.411	0.758	0.591	0.587	44.30	51.00	52.50	49.27	-13.00	-8.80	4.60	-5.73
LSTM(OHLC+TI)	0.407	0.759	0.590	**0.585**	55.30	49.40	**55.30**	**53.33**	10.50	-11.20	14.40	4.57
BERT	0.407	0.763	0.596	0.589	**55.70**	51.00	50.80	52.50	**13.00**	-10.70	-1.40	0.30
BERT + LSTM TI	0.421	0.759	0.588	0.589	48.00	48.20	52.90	49.70	-4.40	1.40	20.20	5.73
HIERARCHY	**0.407**	0.764	0.591	0.587	52.90	49.40	51.20	51.17	8.00	-11.20	12.90	3.23
HIERARCHY + LSTM TI	0.431	**0.755**	**0.583**	0.590	47.50	**52.20**	54.10	51.27	-2.30	**12.40**	**20.30**	**10.13**

4.3.2 Hit profit

This our proposed metric to evaluate the profit according to the predicted market direction. It is the summation of the absolute "hit" returns minus the absolute "miss" returns. We propose this as the primary evaluation metric because this profit metric is independent of any trading strategy and could measure the profit magnitude of the correct answer (fee not included yet).

$$Hit\ profit = \sum_{t=1}^{t=n}(2h_t - 1)\, y_t \qquad (3)$$

where h_t, n, and y_t are described in 4.3.1.

4.4 Implementation Details

We optimized the deep learning model using the Adaptive Moment Estimation (Adam) algorithm and mean square error as loss function for the regression task. The hyperparameters are as follows; dropout probability (0.0, 0.5); BERT aggregations (sum, avg, max); the numerical feature window size is five days and hidden unit (16,64). The hierarchical embedding hidden unit required the hidden unit equals the pre-trained word2vec embedding, which is 300 in this paper (thai2fit) [13]. We test the model with the best validation loss on the best validating epoch.

5. RESULTS AND DISCUSSION

In this section, we discuss the experimental results on four main topics; the overview comparison, the effects of numerical and textual features, the effectiveness of the textual representation approach, finally, a discussion on model interpretability. Table 3 shows the performance comparison of three evaluation metrics. On the RMSE, all deep learning models produce indifferent error around 0.58-0.59%. Next, On the accuracy, the LSTM with numerical features (OHLC + TI) delivers highest at 53.4%. Our model (HIERARCHY/BERT + LSTM TI) performs not so well on the trend accuracy. We could argue that the accuracy metric did not translate directly to more profit; for example, we investigate the BERT and LSTM OHLC then found that the model only predicts one class either all negative or all positive trends. These reflect on the hit profit metric, both BERT and LSTM OHLC show less profit even the BERT has accuracy at 52.5%. On the other hand, for hit profit, our proposed model significantly outperforms other models (HIERARCHY/BERT + LSTM TI), at 10.1% and 5.8% hit profit.

5.1 Effects of Numerical and Textual Features

The performance shows that OHLC features are not sufficient. The LSTM(OHLC) shows poor performance at only -5.8% hit profit. The technical indicator improves the hit profit to 4.6%. Using only textual features (BERT and HIERARCHY) could not succeed with the LSTM(OHCL+TI). On the other hand, we found a significant hit profit gain when we combined both types surpassing the LSTM(OHLC+TI). The last two rows of Table 4 column "TEXT+TI" show that adding numerical features over textual model improves hit profit by +5.5%, +6.9% for the BERT, and HIERARCHY, respectively.

Table 4. Input type effect on hit profit (three-year average).

Type	Model	Hit profit	Compare OHLC+TI	Text +TI
Numeric	LSTM(OHLC+TI)	4.6%	-	-
Textual	BERT	0.3%	-4.3%	-
	HIERARCHY	3.2%	-1.4%	-
Numeric + Textual	BERT + TI	5.8%	+1.2%	+5.5%
	HIERARCHY + TI	**10.1%**	**+5.6%**	**+6.9%**

Table 5. Representation effectiveness (three-year average).

Representation	Model	Hit profit	Difference
BERT	BERT	0.3%	-
	BERT + TI	5.8%	-
HIERARCHY	HIERARCHY	3.2%	+3.0%
	HIERARCHY + TI	**10.1%**	**+4.4%**

5.2 Effectiveness of Textual Representation Approaches

On the effectiveness of the textual representation approach, we found that the HIERARCHY shows better results than BERT at around 3.0-4.4%. Table 5 shows the comparison results. This improvement, maybe because of the weights in the embedding layer of the HIERARCHY, is optimized and updated during the training process. Meanwhile, the BERT has static representation inputs, as described in section 3.3.

5.3 Explainability and Model Interpretation

Initially, we select the hierarchical neural network representation[1] as our priority implementation due to its visual explainability, as illustrates in the original paper [8]. The author utilizes the Layer-

[1] Our BERT approach is a character-based model with no word or sentence level, thus not suitable for model interpretability.

wise relevance propagation (LRP)[15] to leverage model insights on textual news inputs. However, the implementation of LRP to the LSTM is limited on our model because of the required backpropagation chain-rule and we introduce the combination of textual and numeric features. Alternatively, we applied the Integrated Gradient [10] attribution methods within the Captum.ai framework to our model. The Integrated Gradients is an axiomatic model interpretability algorithm that satisfies sensitivity and implementation invariance, which is better than LRP and is simpler to implement. The Layer Integrated Gradients can track the features attribution within neural network layers. So, we select in the word embedding and sentence embedding layer in the hierarchical model to visualize each textual input. We summarized all word importance value (thousand scale) and lists out the top 10 positive and negative words (contributes to return% prediction) during the testing period as shown in Table 6.

Interestingly, we could extract some keywords, symbols, or tokens that, in general, the model observes contribution to the prediction. In our opinion, model interpretability has two mains benefits; first, it allows humans to investigate the underlying insight; second, it is also a debugging tool to assist deep learning model development. For example, if the model highlights a symbol like the "@" or "&" which does not make much sense, we could go back to the cleaning process, removing them, reducing noise to the data. However, in this paper, we still lack the qualitative measurement of this model interpretability, such as having domain experts to help verify those textual insights.

6. CONCLUSION

In this research, we improve the performance of the SET index prediction using textual representation and numerical time-series as inputs to the deep learning model. We explore the impact of input types, either textual, numeric, or both of them. The results show that combining two kinds of features could significantly improve model performance up to 5% profit (on the proposed hit profit metric). Next, we found that using the hierarchical neural network for textual representation yield better performance than the BERT aggregated embedding method at around 3-4%. The best model is the Hierarchical neural network with technical indicators showing hit profit at 10.1% (three-year average). Moreover, we explore the interpretability of the textual part to get a glimpse of the deep learning model insights as well as improving them using the integrated gradient method. However, we still lack the qualitative measures of the model interpretability in our paper.

Table 6. Top positive and negative words on test data.

(+)	English def.	Imp.	(-)	English def.	Imp.
%	[symbol]	4.1	-	[symbol]	-7.6
เพิ่ม	[v] to increase [adj] more	3.2	ไม่	[v, adj] no/negative	-2.7
หุ้น	[n] share, stock	2.7	ปี	[n] year	-2.0
ซื้อ	[v] to buy, purchase	2.6	นี้	[adj] now/this	-1.4
ลด	[verb] to reduce	2.0	รับ	[v] receive, support	-1.4
ไทย	[adj, noun] Thai	1.5	ขึ้น	[v, ad] raise; up	-1.6
ขาย	[v] to sell	1.4	โต	[adj] big; large	-1.2
ราคา	[n] price	1.4	รัฐ	[n] government	-0.8
ส่งออก	[v] to export, [noun] exports	1.2	อี	[unknown], letter "E"	-0.7
!	[symbol]	1.2	มี	[v] to possess	-0.6

7. ACKNOWLEDGMENTS

Our work is a part of the research scholarship competition by Capital Market Research Institute (CMRI), The Stock Exchange of Thailand (SET).

8. REFERENCES

[1] Pasupa, K. and Ayutthaya, T.S.N., 2019. Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. *Sustainable Cities and Society 50*, 101615.

[2] Sanboon, T., Keatruangkamala, K., and Jaiyen, S., 2019. A Deep Learning Model for Predicting Buy and Sell Recommendations in Stock Exchange of Thailand using Long Short-Term Memory. In *2019 ICCCS* IEEE, 757-760.

[3] Akita, R., Yoshihara, A., Matsubara, T., and Uehara, K., 2016. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th ICIS*, 1-6.

[4] Vargas, M.R., Lima, B.S.L.P.d., and Evsukoff, A.G., 2017. Deep learning for stock market prediction from financial news articles. In *2017 IEEE CIVEMSA*, 60-65.

[5] Oncharoen, P. and Vateekul, P., 2018. Deep learning for stock market prediction using event embedding and technical indicators. In *2018 5th ICAICTA* IEEE, 19-24.

[6] Ding, X., Zhang, Y., Liu, T., and Duan, J., 2015. Deep learning for event-driven stock prediction. In *IJCAI International Joint Conference on Artificial Intelligence*, 2327-2333.

[7] Haruechaiyasak, C., Kongyoung, S., and Dailey, M., 2008. A comparative study on thai word segmentation approaches. In *2008 5th ECTI-CON* IEEE, 125-128.

[8] Shi, L., Teng, Z., Wang, L., Zhang, Y., and Binder, A., 2018. DeepClue: visual interpretation of text-based deep stock prediction. *IEEE Transactions on Knowledge and Data Engineering 31*, 6, 1094-1108.

[9] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[10] Sundararajan, M., Taly, A., and Yan, Q., 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* JMLR. org, 3319-3328.

[11] Araci, D., 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *arXiv preprint arXiv:1908.10063*.

[12] Padial, D.L., 2019. Technical Analysis Library using Pandas and Numpy GitHub.

[13] pyThaiNLP, 2019. thai2fit ULMfit Language Modelling(21 Aug 2019).

[14] Liu, G. and Wang, X., 2018. A numerical-based attention method for stock market prediction with dual information. *Ieee Access 7*, 7357-7367.

[15] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W., 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one 10*, 7.