

Enhancing detection of zero-day phishing email attacks in the Indonesian language using deep learning algorithms

Yasinta Roesmiatun Purnamadewi, Amalia Zahra

Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received May 30, 2024

Revised Sep 5, 2024

Accepted Sep 28, 2024

Keywords:

Deep learning

FastText

Indonesian bidirectional encoder representation of transformers

Phishing email

Text classification

ABSTRACT

Email phishing is a manipulative technique aimed at compromising information security and user privacy. To overcome the limitations of traditional detection methods, such as blacklists, this research proposes a phishing detection model that leverages natural language processing (NLP) and deep learning technologies to analyze Indonesian email headers. The primary objective is to more efficiently detect zero-day phishing attacks by focusing on the unique linguistic and cultural context of the Indonesian language. This enables the development of models capable of recognizing phishing attack patterns that differ from those in other language contexts. Four models are tested, combining Indonesian bidirectional encoder representation of transformers (IndoBERT) and FastText feature extraction techniques with convolutional neural network (CNN) and long short-term memory (LSTM) deep learning algorithms. The results indicate that the combination of FastText and CNN achieved the highest performance in accuracy, precision, and F1-score metrics, each at 98.4375%. Meanwhile, the FastText model with LSTM showed the best performance in recall, with a score of 98.9583%. The research suggests exploring deeper into email content or integrating analysis between headers and email content in future studies to further improve accuracy and effectiveness in phishing email detection.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Yasinta Roesmiatun Purnamadewi

Department of Computer Science, BINUS Graduate Program-Master of Computer Science

Bina Nusantara University

St. Raya Kb. Jeruk No.27, RT.1/RW.9, Kb. Jeruk, Kec. Kb. Jeruk, Jakarta, Indonesia

Email: yasinta.purnamadewi@binus.ac.id

1. INTRODUCTION

Phishing is a special form of spam, which utilizes two strategies, namely through social engineering and embedding links in emails that direct victims to fake websites [1]. Email phishing is a social engineering technique that uses fake emails or those that appear to come from trusted sources to lure victims into revealing sensitive information and data and being directed to certain links. Social engineering is a cyber threat that exploits human weaknesses, involving social manipulation and psychological influence to unlawfully obtain sensitive information. One common form of social engineering is phishing, which aims to lure victims into providing personal information without realizing it [2]. Phishing is one of the most impactful threats to enterprises [3]. Detection of phishing emails is crucial in protecting sensitive data and maintaining information security. Every day, organizations are faced with significant challenges when links from phishing emails are clicked by users, even if only by one user [4]. Zero-day attacks are attacks that involve using hosts that are not blacklisted, or utilizing techniques that circumvent the usual approaches to phishing

detection [5]. Although various phishing email detectors have been invented, the challenges still remain that require solutions that can adapt to human intelligence, as attackers continue to develop and modify their phishing models [6].

Bagui *et al.* [7] conducted a study that examined text content in the email body to detect phishing emails. This study evaluated accuracy and computation time. They tested several models using machine learning algorithms such as naïve Bayes, support vector machine, and decision tree, as well as deep learning algorithms such as long short-term memory (LSTM) and convolutional neural network (CNN). They used two text representation methods, namely one-hot encoding and word embedding, specifically continuous bag of word (CBOW). In this research, word embedding is only applied to CNN. The test results show that CNN with word embedding achieves the highest accuracy rate of 96.34%, while Naïve Bayes with one-hot encoding shows the lowest computation time effectiveness.

Somesha and Pais [8] conducted a study to identify phishing emails by focusing on four features of the email header, namely: from, return-path, subject, and message-id. They used various algorithms including random forest, decision tree, support vector machine, XGBoost, and logistic regression. The test results showed that RF achieved the highest accuracy of 99.50%, using FastText (CBOW) on the first of the three datasets tested. In addition, random forest showed good performance consistency with all word embedding algorithms.

Alhogail and Alsabih [6] conducted research to detect phishing emails by analyzing the body of the email text. Graph convolutional network algorithm is used in this research. Based on the evaluation results, the accuracy rate is 98.2% and the false positive rate is 0.015.

Alsufyani and Alzahrani [9] conducted research for phishing email detection. Natural language processing (NLP) and machine learning techniques were used in analyzing the email text. Emails that do not use English, hexadecimal messages, html codes in emails, and emails that only contain links are manually deleted and not used in this study. The machine learning algorithms used are K-nearest neighbor (KNN), multinomial naïve bayes (MNB), decision tree, and AdaBoost. From the test results, the KNN, decision tree, and AdaBost algorithms show good accuracy results, while the MNB algorithm does not show great accuracy results.

Fang *et al.* [10] proposed THEMIS, which is a model for detecting phishing emails by analyzing the email headers and body. THEMIS uses a region-based convolutional neural network (R-CNN) model that is enhanced by using bidirectional long short-term memory (BiLSTM). THEMIS was then tested and compared with CNN and LSTM algorithms. From the test results, THEMIS obtained higher accuracy.

Sankhwar *et al.* [11] proposed enhanced malicious URLs detection (EMUD) which detects whether the url in an email is a real or legitimate web address. To distinguish between phishing and legitimate URLs, the EMUD model focuses on 14 important heuristic features, viz: blacklist, number of dots, visual similarity, double slash, port number, domain length/URL length, country code validation, @ symbol, special characters, IP address in URL, ASCII code, hexadecimal, HTTP in domain section, and domain age. From the test results, the EMUD model with support vector machine has better accuracy and the shortest time.

Smadi *et al.* [12] proposed a phishing email detection system (PEDS) that is able to adapt to environmental changes. PEDS is an online phishing email detection model using reinforcement learning method. A new algorithm is developed namely feature evaluation and reduction (FEaR) to explore behavioral changes and rank features. The authors also developed a new classification algorithm called DENNuRL, the core of this model is neural network. Based on the test results, the proposed technique works well and is able to handle zero-day phishing attacks, with an accuracy result of 98.63%.

In this research, deep learning algorithms such as CNN and LSTM are applied to detect zero-day phishing email attacks in the Indonesian language. The main advantage of CNN is that it can be trained without any special features and is efficient in processing multidimensional input data [7]. Whereas LSTM, as a type of RNN architecture, overcomes the problem of long-term information loss [13], allowing the network to remember information over a longer period of time, which is particularly useful for sequential data processing. This approach is expected to improve the efficiency and accuracy of detecting phishing email attacks in the Indonesian language environment. By focusing our research on the Indonesian language, we hope to make specific contributions in a unique linguistic and cultural context. Through a deeper understanding of email header features in Indonesian, we aim to build a model that can recognize phishing attack patterns that may differ from other language contexts. We also seek to build a model that is adaptive and responsive to new tactics implemented by attackers in the Indonesian language environment.

2. METHOD

The header section of an email consists of a series of structured fields that identify certain information about the message, such as the sender, recipient, date, subject, and more. Meanwhile, the body of

the email is the part that is always visible because it contains the actual message that the email is trying to convey [1]. A comparison with the body of the email shows that the header section displays better regularity. When an attacker forges the sender's identity to send an email with the aim of deceiving the victim, some parts of the header cannot be changed [10]. The use of features in the header is expected to increase the accuracy and speed in identifying phishing emails, so that users can be more effective and efficient in managing information security.

The combination of NLP and machine learning has played an important role in detecting phishing emails [10]. NLP is a subset of artificial intelligence that focuses on the development and implementation of systems and algorithms that facilitate interaction with human language [14]. In this study, we propose a phishing detection model in Indonesian using deep learning algorithms such as CNN and LSTM, supported by text extraction features from Indonesian bidirectional encoder representation of transformers (IndoBERT) and FastText. Deep learning approaches have made impressive achievements in various tasks, including in NLP [15]. Deep learning is used due to its powerful computational capabilities to overcome the weaknesses of traditional machine learning methods [16]. As a subset of machine learning, deep learning aims to develop algorithms and computational models that mimic the learning process of the human brain, especially when it comes to processing complex and abstract information. The principle of deep learning is to enable computers to "learn" from data in a similar way to humans, which is achieved through complex networks and adjustable parameters, allowing the models to identify very complicated patterns in the data [17]. The stages of solution implementation in this research are shown in Figure 1.

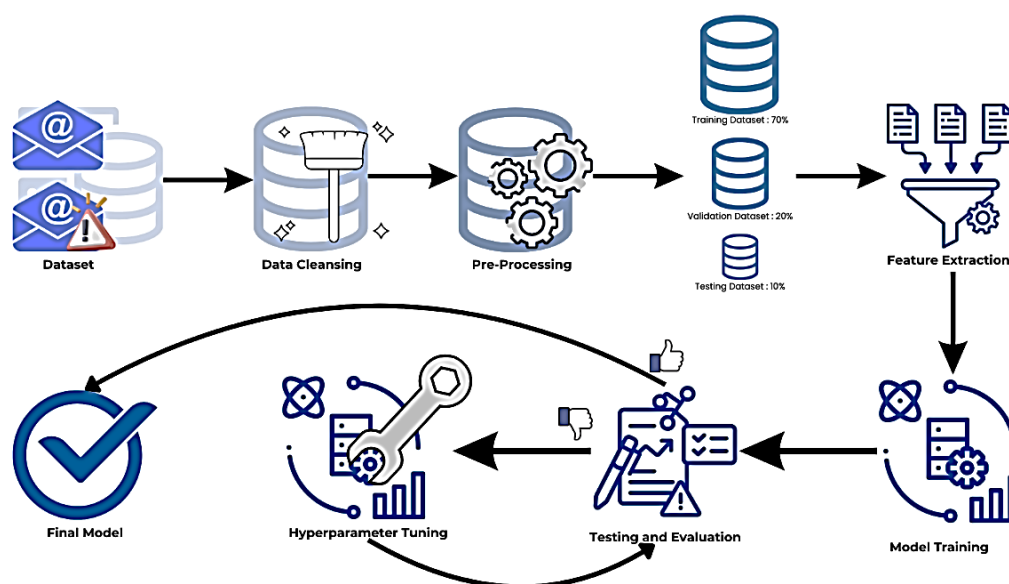


Figure 1. Solution implementation stages

The solution implementation phase begins with data cleaning to remove noise and missing values, followed by data preprocessing to properly prepare the data. Next, the data set is divided into subsets for training, validation, and testing. Selected features are extracted using IndoBERT or FastText models. The classification model is built and trained on the training subset and then tested on the testing subset to evaluate its performance. If the model performance is not satisfactory, hyperparameter tuning is performed and the model is tested again. If the performance is as expected, the model is considered a final model and can be used to predict new data.

2.1. Data collection

This research uses an email dataset from the Indonesian Ministry of Public Works and Housing during the year 2023. The obtained dataset consists of 6,669 emails identified as phishing attempts and 2,726 legitimate emails. In reality, the number of legitimate emails is higher than the number of phishing emails, but due to security reasons, the number of legitimate emails obtained is limited. After data preprocessing, the number of emails used was 3,844, consisting of 1,922 phishing emails and 1,922 legitimate emails. The email part used in this research is the email header part. The header features evaluated include 'Subject', 'From (Header Address)' and 'From (Header Name)'.

2.2. Data preprocessing

The steps implemented in data cleaning are: i) return of empty strings for data that has missing values, ii) removing emails with empty subjects, iii) removing emails whose subjects are written entirely in a foreign language, and iv) the focus of this research is on emails written in Indonesian, so only emails with subjects written entirely or partially in Indonesian are retained. The use of a mixture of Indonesian and foreign languages is retained to maintain the natural impression of Indonesian language use.

To balance the dataset used, 1,922 phishing emails and 1,922 legitimate emails were selected for use in this study, resulting in a total dataset of 3,844 emails used for supervised learning. Data pre-processing is performed, namely:

- Lower case for header characteristics: subject and from (header name).
- Cleansing for header characteristics: subject and from (header name).
- Stemming for header properties: subject.
- The literary library used in this research is a widely recognized library specifically designed for the Indonesian language [18]. This library helps in various operations such as stemming, which aims to identify the basic form of words in Indonesian [19].
- Tokenization for header characteristics: subject, from (header name) and from (header address).

For the subject and from (header name) features, tokenization is performed by breaking the text into words based on spaces. While the from (header address) feature breaks the text into words based on the @, ., and space characters. After data preprocessing, the dataset is divided into 3 parts: i) training dataset: 2.690 (70%), ii) validation dataset: 769 (20%), and iii) test data set: 385 (10%).

2.3. Feature extraction using IndoBERT

IndoBERT is a special variant of BERT for the Indonesian language [20]. It represents an attempt to optimize the understanding and use of language models specifically in the context of the Indonesian language. This model, based on the BERT architecture, has proven to be effective in understanding and generating text in multiple languages [21]. Before entering the text classification stage, the text goes through a pre-training process using the IndoBERT model. This model uses the 768-dimensional vector representation generated by BERT as input features [22].

2.4. Feature extraction using FastText

FastText effectively handles out-of-vocabulary words with the n-gram approach [23], dividing unknown words from the corpus into n-grams that may be similar to words in the vocabulary. The FastText model used was built using the CBOW technique, considering position weights, 300 dimensions, and using n-gram characters of 5, window size 5, and 10 negative values [24].

2.5. Text classification

The reference for CNN and LSTM architecture configuration and hyperparameters partly follows the research of Bagui *et al.* [7]. The CNN and LSTM models were built using Adam's optimizer, binary cross entropy loss function, and rectified linear unit (ReLU) activation function was used except for the dense layer which uses sigmoid. The combination of functions proved to be effective in handling binary data [7], [16] where 1 represents phishing emails and 0 represents legitimate emails. The architecture of the CNN and LSTM models is as follows:

a. CNN modeling

The CNN architecture configuration used in this research:

- Input : number of tokens: 60, dimensions: IndoBERT or FastText custom dimensions.
- Convolutional 1 : filter: 32, kernel: 5, Stride:1, activation: ReLu.
- Max-Pooling 1 : pool size: 2.
- Convolutional 2 : filter: 64, kernel: 2, stride:1, activation: ReLu.
- Max-Pooling 2 : spatial size: 2.
- Flatten
- Dense : activation: ReLu.
- Output : activation: sigmoid, neuron: 1 neuron for binary classification.

b. LSTM modeling

The LSTM architecture configuration used in this research:

- Input : number of tokens: 60, dimensions: IndoBERT or FastText custom dimensions.
- LSTM 1
- LSTM 2
- Flatten

- Dense : activation: ReLu.
- Output : activation: sigmoid, neuron: 1 neuron for binary classification.

2.6. Hyperparameter

Hyperparameter tuning is performed using the grid search method and 5-fold cross-validation. This means that the dataset will be divided into 5 parts, where the model will be trained and validated 5 times, with each part used as validation data at different iterations. This process will use GridSearchCV from Scikit-learn, which uses cross-validation to evaluate estimator performance and select the best parameters. GridSearchCV also supports the distribution of computation across multiple cores, speeding up the parameter selection process through simultaneous computation [25]. The hyperparameters to be used in CNN or LSTM models are:

- Hyperparameters of CNN model: the hyperparameters for the CNN model are: epochs set to 25, kernel sizes of 3, 5, 7, 9, and 11, and pooling sizes of 2, 3, 4, and 5.
- Hyperparameters of LSTM model: the hyperparameters for the LSTM model are: epochs set to 25 and hidden node sizes of 4, 8, 16, 32, 64, and 128.

2.7. Evaluation

Model performance in this research will be measured using accuracy, precision, recall, and F1-score. These matrices are commonly used in text classification [26]. The formula is as (1)-(4):

- Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Precision

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

- Recall

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

- F1-score

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

The confusion matrix provides the basis for the calculation of these metrics by providing information about the number of correct and incorrect predictions made by the model in each class or category.

		Predicted	
		N	P
Actual	N	TN	FP <i>(Error Type I)</i>
	P	FN <i>(Error Type II)</i>	TP

(5)

Negative value=represents clean email

Positive value=represents phishing email

3. RESULTS AND DISCUSSION

This research uses Python as the programming language. Jupyter Notebook runs on Google Colab Pro as an integrated development environment (IDE) [27]. This provides access to more powerful resources, as deep learning requires high computational resources [28], thus speeding up the experiment significantly. Based on the search results for the best hyperparameters using grid search, the highest performance in accuracy, precision, and F1-score matrices was achieved by the FastText and CNN model with a configuration of 15 epochs, a kernel size of 11, and a pooling size of 2. Meanwhile, the best performance in

the recall matrix was achieved by the FastText and LSTM model with a configuration of 25 epochs and 32 LSTM units. The results of the tested models are shown in Table 1.

Table 1. Performance comparison

Matrix	IndoBERT and CNN	IndoBERT and LSTM	FastText and CNN	FastText and LSTM
Accuracy (%)	97.3958	97.1354	98.4375	97.6563
Precision (%)	98.4043	97.8836	98.4375	96.4467
Recall (%)	96.3542	96.3542	98.4375	98.9583
F1-Score (%)	97.3684	97.1129	98.4375	97.6864

Based on the Table 1, various model combinations have been analyzed for detecting phishing emails. The FastText and CNN combination demonstrated the best performance among all models, achieving the highest accuracy, precision, and F1-score, each at 98.4375%. This indicates that this model is highly effective and consistent in correctly identifying phishing emails. The FastText and LSTM combination also showed very good performance, with the highest recall value of 98.9583%, indicating its ability to detect almost all phishing emails present. Although its accuracy, precision, and F1-Score are slightly lower compared to FastText and CNN, this model remains competitive with an accuracy of 97.6563% and an F1-score of 97.6864%.

On the other hand, the IndoBERT and CNN and IndoBERT and LSTM combinations also demonstrated good performance, although they fall below the performance of FastText-based models. IndoBERT and CNN achieved an accuracy of 97.3958% and an F1-Score of 97.3684%, while IndoBERT and LSTM achieved an accuracy of 97.1354% and an F1-score of 97.1129%. Both combinations have the same recall value of 96.3542%, but the precision of IndoBERT and CNN is slightly higher compared to IndoBERT and LSTM.

Overall, this table shows that FastText-based models excel in phishing email detection. The FastText and CNN combination provides the best performance in terms of accuracy, precision, and F1-score, while the FastText and LSTM combination excels in recall. Although IndoBERT-based models also show good performance, FastText-based models are generally more effective in detecting phishing emails, with FastText and CNN being the most consistent across key metrics. The confusion matrix for these models can be seen in Figure 2.

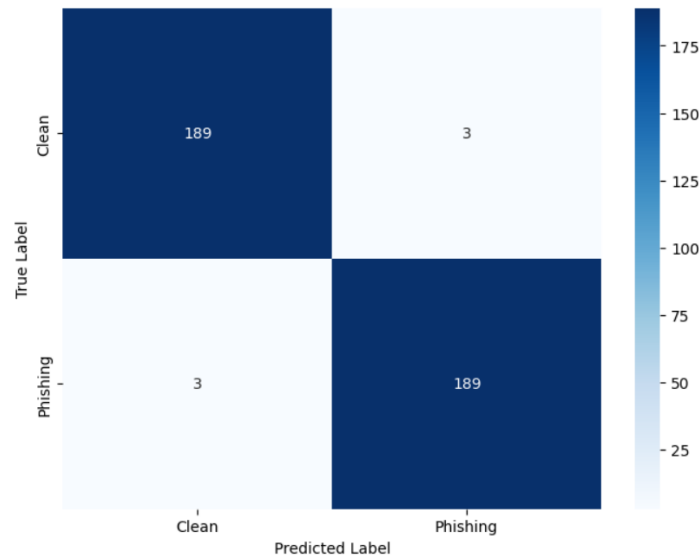


Figure 2. Confusion matrix

This matrix indicates that the model performs exceptionally well, correctly identifying 189 clean emails and 189 phishing emails. However, there are a few errors where 3 clean emails were incorrectly classified as phishing (false positives), and 3 phishing emails were mistakenly identified as clean (false negatives). With such a small number of errors compared to the number of correct predictions, this matrix

underscores that the model has high accuracy and is well-balanced in detecting phishing emails and distinguishing between harmful and safe emails.

4. CONCLUSION

This research successfully integrates FastText and IndoBERT word embedding techniques with CNN and LSTM deep learning architectures to detect zero-day phishing email attacks in Indonesian, focusing on email header features. The results demonstrate that the FastText and CNN combination achieves the highest accuracy, precision, and F1-score, indicating excellent performance in phishing detection. The FastText and LSTM combination also performs very well, particularly in recall, which is crucial for identifying as many phishing cases as possible. While slightly lower, the IndoBERT, CNN, IndoBERT, and LSTM combinations still deliver reliable results and are viable alternatives.

Overall, using FastText with CNN or LSTM proves highly effective for detecting phishing in Indonesian emails, with IndoBERT and CNN serving as a strong alternative. However, this research has limitations, as it only analyzes email headers. For future research, it is recommended to expand the scope of analysis by including email body features, which allows more sophisticated NLP techniques to detect phishing indicators in more detail. Additionally, improving parameter tuning in deep learning architectures and implementing the latest algorithms that capture complex feature relationships, along with expanding the dataset to include various email formats, will enhance model accuracy and robustness.

ACKNOWLEDGEMENTS

This research paper is part of the master of informatics engineering thesis at Bina Nusantara University. The author would like to express sincere gratitude to all those who have made valuable contributions to the completion of this paper. Special thanks are extended to Bina Nusantara University for their financial support of this research. It is hoped that this effort will not only be useful but also serve as a steppingstone for future research.





REFERENCES

- [1] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 2070–2090, 2013, doi: 10.1109/SURV.2013.030713.00020.
- [2] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Computing*, vol. 25, no. 6, pp. 3819–3828, Dec. 2022, doi: 10.1007/s10586-022-03604-4.
- [3] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommunication Systems*, vol. 76, pp. 139–154, 2021, doi: 10.1007/s11235-020-00733-2.
- [4] S. Phomkeona and K. Okamura, "Zero-day malicious email investigation and detection using features with deep-learning approach," *Journal of Information Processing*, vol. 28, pp. 222–229, 2020, doi: 10.2197/ipsjip.28.222.
- [5] D. L. Cook, V. K. Gurbani, and M. Daniluk, "Phishwish: a simple and stateless phishing filter," *Security and Communication Networks*, vol. 2, no. 1, pp. 29–43, 2009, doi: 10.1002/sec.45.
- [6] A. Alhogaib and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security*, vol. 110, p.102414, Nov. 2021, doi: 10.1016/j.cose.2021.102414.
- [7] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine learning and deep learning for phishing email classification using one-hot encoding," *Journal of Computer Science*, vol. 17, no. 7, pp. 610–623, 2021, doi: 10.3844/jcssp.2021.610.623.
- [8] M. Somesha and A. R. Pais, "Classification of phishing email using word embedding and machine learning techniques," *Journal of Cyber Security and Mobility*, vol. 11, no. 3, pp. 279–320, 2022, doi: 10.13052/jcsm2245-1439.1131.
- [9] A. A. Alsufyani and S. M. Alzahrani, "Social engineering attack detection using machine learning: Text phishing attack," *Indian Journal of Computer Science and Engineering*, vol. 12, no. 3, pp. 743–751, May 2021, doi: 10.21817/indjcs/2021/v12i3/211203298.
- [10] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *IEEE Access*, vol. 7, pp. 56329–56340, 2019, doi: 10.1109/ACCESS.2019.2913705.
- [11] S. Sankhwar, D. Pandey, and R. A. Khan, "Email phishing: an enhanced classification model to detect malicious URLs," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 6, no. 21, 2019, doi: 10.4108/eai.13-7-2018.158529.
- [12] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decision Support Systems*, vol. 107, pp. 88–102, Mar. 2018, doi: 10.1016/j.dss.2018.01.001.
- [13] L. F. Simanjuntak, R. Mahendra, and E. Yulianti, "We know you are living in bali: location prediction of twitter users using BERT language model," *Big Data and Cognitive Computing*, vol. 6, no. 3, Sep. 2022, doi: 10.3390/bdcc6030077.
- [14] I. Lauriola, A. Lavelli, and F. Aiolli, "an introduction to deep learning in natural language processing: models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022, doi: 10.1016/j.neucom.2021.05.103.
- [15] J. Ahmad, H. Farman, and Z. Jan, "Deep learning methods and applications," in *SpringerBriefs in Computer Science*, Springer, 2019, pp. 31–42. doi: 10.1007/978-981-13-3459-7_3.
- [16] R. Alotaibi, I. Al-Turaiki, and F. Alakeel, "Mitigating email phishing attacks using convolutional neural networks," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, Mar. 2020, pp. 1–6, doi: 10.1109/ICCAIS48893.2020.9096821.
- [17] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," May 27, 2015, *Nature Publishing Group*, doi: 10.1038/nature14539.





- [18] N. Yusliani, R. Primartha, and M. Diana, "Multiprocessing stemming: a case study of Indonesian stemming," *International Journal of Computer Applications*, vol. 182, no. 40, pp. 15–19, Feb. 2019, doi: 10.5120/ijca2019918476.
- [19] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving text preprocessing for student complaint document classification using sastrawi," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jul. 2020, vol. 874, no. 1, p. 012017, doi: 10.1088/1757-899X/874/1/012017.
- [20] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: a benchmark dataset and pre-trained language model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, Stroudsburg, PA, USA: International Committee on Computational Linguistics, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
- [21] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, Jun. 2017.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for Language understanding," in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
- [23] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, Dec. 2017, doi: 10.1162/tacl_a_00051.
- [24] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv*, Feb. 2018, doi: 10.48550/arXiv.1802.06893.
- [25] F. Pedregosa *et al.*, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [26] Q. Li *et al.*, "A survey on text classification: from shallow to deep learning," *arXiv*, 2020, doi: 10.48550/arXiv.2008.00364.
- [27] F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.
- [28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

BIOGRAPHIES OF AUTHORS



Yasinta Roesmiatun Purnamadewi     is currently a graduate student in Computer Science at Bina Nusantara University, Indonesia. She received her associate's degree in 2009 and bachelor's degree in 2012 in Computer Science from STMIK AMIKOM Yogyakarta. She can be contacted at email: yasinta.purnamadewi@binus.ac.id.



Amalia Zahra     is a lecturer at the Master of Information Technology, Bina Nusantara University, Indonesia. She received her Bachelor's degree in Computer Science from the Faculty of Computer Science, University of Indonesia (UI) in 2008. She does not have a master's degree. Her Ph.D. was obtained from the School of Computer Science and Informatics, University College Dublin (UCD), Ireland in 2014. Her research interests cover various fields in speech technology, such as speech recognition, spoken language identification, speaker verification, speech emotion recognition, and so on. Additionally, she also has an interest in NLP, computational linguistics, machine learning, and artificial intelligence. She can be contacted at email: amalia.zahra@binus.edu.