Spam Detection in Indonesian Emails: A Comparative Study of SVM and LSTM Using Word2Vec and TF-IDF

Jordan Josdaana, Vianna Calista Tamsila, Rafaell Widjayaa, Juan Kennetha, Jason Timothya

*aComputer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia*

**Abstract**

E-mail is a vital medium of communication in today's rolling cyber era, it deals significantly with various issues, from the personal level of communication to official interaction. At present, spam is the challenging problem which quickly inboxes not only with unsolicited information but also with serious threats, such as malware and phishing attacks. There are many records for Indonesia, evidencing the number of unwanted mails being at a higher rate. In this paper, a new approach to enhancing email spam using advanced machine learning techniques using an SVM-LSTM hybrid scheme is proposed to automatically categorize spam and non-spam emails. The DDL process developed does include a series of preprocessing techniques, like data visualization, tokenization, sentiment analysis, followed by model training and model evaluation using the LSTM-SVM with proper feature set. Experimental results show that the developed approach is effective and outperforms almost all state-of-the-art methods, with the highest accuracy of 97.98% achieved by the LSTM model using TF-IDF. The detailed comparative analysis of various word embedding and kernel methods provides a proper understanding of the importance of the latest strategies in spam detection, especially in Indonesian contexts. This study contributes to further improving the spam detection systems and will have broad implications for enhancing the security of e-mails and reliability of communications both in the Indonesian context and globally.

© 2024 The Authors. Published by ELSEVIER B.V.  
 This is an open access article under the CC BY-NC-ND license ([https://creativecommons.org/licenses/by-nc-nd/4.0](https://nam11.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcreativecommons.org%2Flicenses%2Fby-nc-nd%2F4.0&data=05%7C01%7CPROCS%40elsevier.com%7Ce17ab5ee9c2a4464746408db199d4e81%7C9274ee3f94254109a27f9fb15c10675d%7C0%7C0%7C638131935656073248%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=wfTY5%2FiSCZNT%2F6jJxRNGOtytYpQyG3D6Txuzlb8IwgE%3D&reserved=0))

*Keywords: E-mail, Classification, Spam Detection, SVM, LSTM, NLP*

**1.** **Introduction**

Email today serves, basically, for everyone as a tool of communication in a changing world of digital content: from a business conversation to a personal letter. But email's ever-growing ubiquity and ease of use have deteriorated over the years due to the ongoing problem of spam email. Thus, aside from the huge aggravation that clogging up our inboxes with dozens of unsolicited email messages causes many people, there are some very dangerous security risks here, too-particularly the risks of malware and phishing attacks.

E-mail spam is more than just a nuisance that floods our mailboxes with unwanted messages. It is, in fact, a real cybersecurity threat. Spam emails are loaded with phishing scams, malware, and other types of cyber risks (Pedersen & Brincker, 2021). They can result in huge financial loss and leakages of personal information. In Indonesia, spam email traffic has been growing every year, which makes it an urgent problem that must be addressed with proper solutions.

This problem forms the background in Indonesia and is judged to be one of the worst spam email-related problems, according to a new report. Traditional spam filters are being overwhelmed by the sheer quantity and growing sophistication of spam emails that will invite more complex and sophisticated spam filters (Pamungkas, n.d.). Spam detection mechanisms need to be developed in a more efficient and effective way. The new approach proposed for the solution to the problem of email spams in Indonesia with updated machine learning techniques.

Since machine learning can learn from data and make predictions, it offers some very promising solutions to the problem of e-mail spam. In this regard, this paper shall seek to interrogate the application of two specific machine learning techniques, Support Vector Machines and Long Short-Term Memory networks in the detection and prevention of spam emails in Indonesia (Shaik et al., 2023).

Support Vector Machines, or SVM, is a strong machine learning model for classification and regression analysis. It is especially known for its effectiveness in high-dimensional spaces. This makes this method very suitable for text data as emails (Cortes & Vapnik, 1995).

SVM works by mapping input vectors to a feature space where a hyperplane can be used to perform classification. This hyperplane is chosen to maximize the margin between different classes in the training data. That is, it minimizes the norm of the weight vector subject to constraints that force the SVM to make its decisions based upon the vectors that are hardest to classify. It is this aspect of the algorithm that makes the SVM robust to overfitting (Cortes & Vapnik, 1995)(John-Africa & Emmah, 2022).

Now, those are things that SVM does extremely well: dealing with non-linear decision boundaries by way of the Kernel Trick. The kernel trick allows the SVM to work in the original input space rather than having to compute the coordinates of the data in that high-dimensional space (Cortes & Vapnik, 1995)(Shaik et al., 2023). Samantha Sloan and Jason Morton, in their book titledPython Machine Learning, wrote: “It makes the SVM not only effective but also computationally efficient”. They are a class of recurrent neural networks with a capacity for learning their long-term dependencies. It makes them particularly applicable to sequential data such as e-mails.

One of the main problems of traditional RNNs is that of the vanishing gradient problem. In this type of network the contribution of information geometrically decays over time. This flaw is overcome in the LSTM due to its unique design. Moving towards discernment an LSTM unit consists of memory cell and three multiplicative units; input gate, output gate and forget gate. All these three kinds of gates control the flow of information into, out of and within the cell respectively (Cortes & Vapnik, 1995).

What makes LSTMs really effective in sequential data-based tasks is its ability to learn as well as recall and retrieve information from a long period of time (Shaik et al., 2023). When it comes to spam e-mail detection utilizing this, what the LSTM becomes capable of learning is not only through email content, but even through the sequence of terms or phrases that appear in it (Shaik et al., 2023). The aim of this project is to bring about increased accuracy and higher-speed capabilities in spam detection, ultimately leading to a better and quick clean email experience within Indonesia. The finding of this project could be very useful in the real world and it will also help others in gaining experience through its study of spam detection in machine learning.

**2. Related Works**

The identification and elimination of spam in emails have been central to the progress of machine learning and natural language processing over a number of years. Multiple attempts have been made to invent, develop and evaluate various techniques and models which can enhance accuracy and efficiency in the spam detection system. In this part, some prominent works on email spam detection are reviewed, concerning approaches used in machine learning and deep learning techniques as well as their application towards Indonesian emails.

2.1 Machine Learning Approaches to Spam Detection

Due to their ability of handling text classification tasks effectively, traditional machine learning methods are often used for spam detection. Some of the widely utilized algorithms include Naïve Bayes, Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbors (KNN).

* Naïve Bayes: One of the initial classical approaches that were employed for spam detection is Naïve Bayes classifier; it uses Bayes’ theorem conceptually with independence assumption among features. Owing to its simplicity and effectiveness, it has become popularly used as a method for filtering out junk mails. (Androutsopoulos et al., 2000) showed how good Naïve Bayes is when applied in detecting spams at high rates with reasonable computational expenses. from the previous research done to classify Indonesian spam email detection, most of them use Naïve Bayes (Hidayat Informatika, 2023) (Anugroho et al., 2005).
* ​​Support vector machines (SVM): Robustness and high accuracy in spam detection have resulted in widespread use of SVMs. Drucker et al. (1999) investigated the possibility of using SVMs for spam filtering, illustrating the importance of dealing with large numbers of attributes and their efficiency in text classification tasks. (Sahami et al., 1998) showed that combining SVMs with other methods such as feature selection and ensemble methods can further enhance performance.
* Decision Trees and Random Forests: On the other hand, Decision Trees and their ensemble variant, Random Forests, have also been applied to spam detection. In particular, their interpretability and capability to deal with non-linear relationships render them appropriate for this task. In this regard, (Zhang et al., 2004) suggested a combination of Decision Trees with boosting techniques for building a model to detect spams leading to notable improvements in accuracy levels. This model also used in classifying spam in that is written in Malaysian, similar to Indonesia (Abdulrahman & Salim, 2022).

2.2 Deep Learning Techniques

The successes of deep learning have been followed by a slew of newer exploits, spam detection included. Deep learning models, especially the Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), have been mostly outstandingly applied in text classifications tasks throughout.

* Recurrent Neural Networks (RNNs): RNNs, in specific Long Short-Term Memory (LSTM) networks, have been popularly used for spam detection task as they can capture long dependencies in sequential data. Although the work of (Hochreiter & Schmidhuber, 1997) was the pioneering work for the development of LSTMs, its application in most text classification tasks, including spam detection tasks, has been pronounced. For instance, (Dada et al., 2019) were among those who employed LSTM networks in email spams classification, where the model outperformed the performance of traditional machine learning models.
* Convolutional Neural Networks (CNNs): CNNs have also been used for spam detection, especially to detect local patterns in text data (Rahman & Ullah, 2020). One of the initial persuasive innovations of Kim (2014) was the introduction of the CNN in sentence classification. This capability was used for spam detection. This ability—automatically deriving feature representations from raw text data—spells the power that lies beneath spam classification tasks.

2.3 Hybrid Models

Combining different models with the machine learning and deep learning techniques is a popular method, as the objective of these models is to leverage the benefits of each. Studies proved that the hybrid model does well in enhancing the accuracy and robustness of spam data detection.

* Combination of SVM and LSTM: This hybrid model goes beyond the proposed integration of SVM and LSTM and is thus a hybrid model. Of course, SVM is good at dealing with high-dimensional feature spaces, and LSTM (Jain et al., 2019) are credited for their optimum utilization of capturing sequential dependencies in the involved text data. Some of the prior studies on the analogous hybrid model are a study conducted by Islam et al., 2020; such studies have claimed that the proposed hybrid model inspired from them could effectively improve the metrics of the spam detection performance: accuracy and AUC.
* Ensemble Techniques: Previous research studies have attempted to use ensemble methods, namely biological immune-like algorithms, for spam detection comprising multiple spam filters: bagging model, boosting model, stacking model. For example, the ensembling model by Chhabra et al., 2016, combined Naïve Bayes, SVM, and Decision Trees; through this, they reported phenomenal improvements in accuracy. Another research has utilized ensemble learning for improving its accuracy (Adnan et al., 2024).

2.4 Spam Detection for Indonesian Email

The issue of spam detection for Indonesian emails has received even less attention, likely due to newer analyses or smaller databases in the respective problem definition than the previous work in different language settings. However, some of the works motivate the analysis in this specific setting.

* Indonesian Spam Filtering: Prasetyo and Nurjanah (2016), in their work, worked on the issue of spamming Indonesian documents. They experimented with a battery of machine learning techniques to prove challenges against the Indonesian emails corpus.
* Deep Learning for Indonesian Texts: From the above, deep learning approaches have equally found applications in Indonesian document text classification, for example, as was the case in the study by Hartono et al., 2019, where the authors explored the use of LSTM networks for Indonesian sentiment analysis tasks, similarly gaining proofs of the proper usage of deep learning approaches for any Indonesian Language processing task, such as, for spam detection.

Conclusively, the basis for defining a good arsenal system of spam detection for Indonesian emails should depend on the previous literature. While the combination of traditional machine learning, deep learning methods, and hybrid techniques offers promise in this research, the development of a good system for the specific context of spam detection is full of challenges. The combination of SVM and LSTM proposed in this work should also join the ongoing efforts directed at building upon the earlier advancements in the direction of improved email security and communication efficiency in Indonesia.

**3. Methodology**

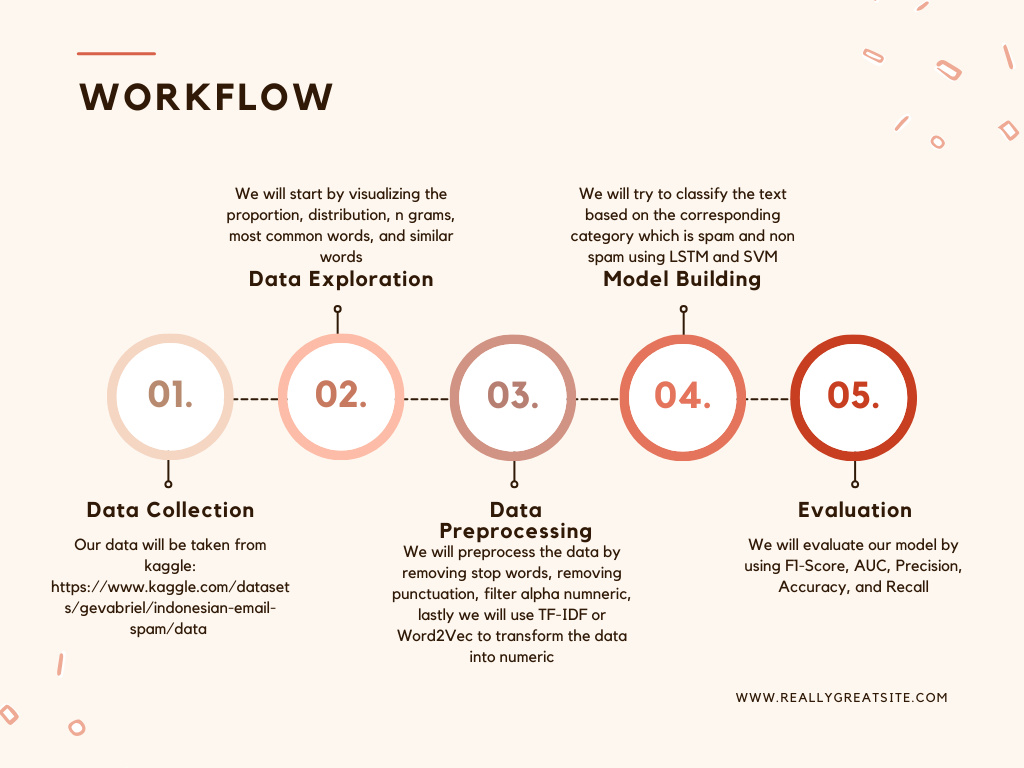
****

Fig. 1. Workflow of the Project

Fig. 1. shows the workflow of the study conducted, the process will include data collection, data exploration, data preprocessing, model building and evaluation. In subsection 1, Data Collection, we gather all relevant information useful for analysis and in the construction of a model. This step is the most critical as this will generate a quality and expansive dataset. Subsection 2, the Data Exploration we visualize data and analyze them to show patterns and trends that can support the discovery of essential features and anomalies. Thus, the Subsection 3, Data Preprocessing, involves data cleaning and making data ready to use and preparing it for modeling to improve the accuracy of the model. Subsection number 4 is Model Building. Here, a model has to be built and trained with the proper algorithm which gives efficient and accurate results. Section number 5, Evaluation, is the part where the model is evaluated based on the metrics result such as F1-Score, AUC, Precision, Accuracy, Recall. It helps to evaluate the model, and its effectiveness and make the necessary adjustments. The details of each process will be analyzed thoroughly in this chapter.

*3.1 Data Collection*

Data was collected from https://www.kaggle.com/datasets/gevabriel/indonesian-email-spam/data. This dataset contains emails in Indonesian categorized as spam or ham, with 52% spam and 48% ham, making a total of 2620 entries. After data retrieval, the data is inspected to understand the structure of the distribution: to check that there are no duplicates, missing values, and to prepare the data for further work on the identification of unique values and the initial stage of data cleaning.

*3.2 Data Exploration*

This study attempted a data understanding process by means of a sequence of data containing insight into the usage of this dataset. First, the class distribution, trying to reveal that the number of spam and ham emails was balanced, was examined. Text length distributions under both spam and ham categories were analyzed with histograms to compare the length pattern between the two classes. The word frequency distribution looked at which words occur most often in the dataset, and finally, they created a word cloud where words frequently occurring had more size, resulting in a visually intuitive representation of the dataset's vocabulary. The tips are valuable to collectively understand the characteristics of datasets, which will provide comprehensive insights for preprocessing and model-building tasks in email classifications.

*3.3 Preprocessing*

Data preprocessing begins with loading the dataset into pandas and performing first looks to understand the structure and summary statistics of the dataset. Functions like 'describe', 'head', and 'info' are used to have a clear view of the data. As seen in the figure, a bar plot is used to represent the distribution of categories between spam and ham. Compared to one another, the quantity between the two categories is presented. Furthermore, the comparison in the text length analysis brings out details of the lengths of a single message and plots histograms for spam and ham classes. Therefore, a comparative understanding can be noticed based on the message length.

Notice next that this data cleaning is focused on stripping the dataset of entries for which the 'Pesan' or message column values are absent. Tokenization on text data and stop-word operations are performed aiming to clean the common insignificant words. The most common words of the data set are unearthed and served in the bar plot style, ranging from frequently occurring terms to words. Conversion to numerical format took place using label encoding, and further, text data underwent tokenization or padding. This ensures consistency in the length of the sequences and is an important step to make the data ready for the next machine learning processes.

Advanced preprocessing included the installation of word clouds and n-gram analysis to visualize the frequency of words and common word pairs in the dataset. Sentiment analysis, using TextBlob, compartmentalized the given messages as either Positive, Negative, or Neutral based on the polarity of the sentiment the text carried. Then, the text data is converted to numerical vectors using the TF-IDF vectorizing technique that tends to derive the importance of the words on the basis of frequency and distribution. Finally, the dataset is split into the train and test data, then fitting and testing of an SVM model. Metrics for performance such as accuracy, classification report, and confusion matrix help determine the ability of the model to classify spam from ham.

*3.4 LSTM Model*

The Long Short Term Memory model or the LSTM is a type of recurrent neural network (RNN) that was purposely built for learning from sequences of data, for example in this case is a text. The LSTMs belong to the group of neural networks which are effective in the learning of dependencies in sequential data which are consequently effective in applications like text classification. The LSTM model was built and begin by importing the necessary layers and modules from the TensorFlow library. For the embedding layer, weights will be used that have been pre-trained via a Word2Vec model over a large corpus of text data. With the help of these pre-trained embeddings, which act as the first weights in our embedding layer, words are meaningfully represented according to their corpus context. Moreover, one other version will be tested using Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction. The TF-IDF is a statistical metric that can measure the importance of a word on a document which is connected to the corpus, as this is used to convert text data into a numerical format that can be fed to a machine learning algorithm.

After embedding, the LSTM layer will be added with 100 units. This is done on this layer in order to record complex patterns and dependencies in the successive input data. The LSTM works through mechanisms like the forget gate, input gate, and output gate to maintain and update its cell and hidden states throughout long sequences of the data and prevent the vanishing gradient problem associated with conventional RNNs. After adding the LSTM layer, the next step is to add a dense output layer with one neuron. Then compiled with the configured model using the sigmoid activation function, expecting the fact that the dataset belongs to any of the two categories. The sigmoid function in this case will give the output into a range of 0 to 1. The sigmoid is used to classify the text into spam or ham (not-spam).

Last, for the compilation of the model, the Adam optimizer will be used. This optimizer was chosen as it is one of the best optimizers that could handle sparse gradients on noisy problems. The loss function that has been considered for this binary classification task is binary cross-entropy, which relates to the comparison of the predicted probability and the true binary label. Accuracy will be used as a model evaluation metric on training and validation data. 20 epochs will be used, which indicate an iteration of the model on the whole dataset 20 times, and a batch size of 32, after which the model is updated. In the training-monitoring process, additional splitting of the training data is done into two parts: 80% of the data are for training, while 20% are for validation.

*3.5 SVM Model*

The Support Vector Machine (SVM) model is one of the bases of machine learning and is particularly known for its performance on classification tasks. SVM works by finding an optimal hyperplane inside the feature space that linearly separates the data of different classes. When dealing with text classification problems, the inputs are inherently non-numeric and are sent through preprocessing steps such as the Word2Vec or TF-IDF. These steps allow textual content to be suited for the SVM to handle.

The kernels, namely linear and Radial Basis Function (RBF), are used in the decision-making process of SVM. A linear kernel draws boundaries in straight lines, meanwhile the RBF kernel can identify subtle, non-linear patterns in the data. In the current setting of our task, spam or non-spam classification of tweets, the SVM model will make use of its training on the examples of labeled tweet data to build a decision boundary. This boundary will be used to guide further classification of new, unseen tweets in tune with the learned patterns.

This research classification problem requires to compare the performance of SVM in a scenario where two different word embedding techniques, Word2Vec and TF-IDF, are applied to the dataset. In addition, the research also want to test the effect of the difference in the decision-making mechanism and hence run the SVM using a linear as well as an RBF kernel mechanism. The SVM classifier will be run by utilizing Scikit-Learn. Various kernels of the SVM will be used in order to train with both the Word2Vec and TF-IDF embeddings.

*3.6 Evaluation Metrics*

A standard set of metrics will be used aiming to quantify the predictive performance of the LSTM and SVM models in discriminating among spam and non-spam tweets. Such metrics are objective and refer to measures of classification accuracy and complemented by confusion matrices for detailed visualization. Accuracy, the most important relevant metric, will be used in highlighting an overall idea of the models' behavior and refers to the percentage of accurately classified instances over the total number of instances, specifying, in brief, how accurate the models are in their classification.

As a secondary metric to accuracy, precision measures the capabilities of the models to correctly determine positive instances. The evaluation measures the output of all true positive predictions over the number of instances that are classified as positive, which in other words means that this measures the capability of the models to identify spam tweets correctly. Likewise, recall, or sensitivity, will be the metric of the ability of the models to capture all positive instances over the dataset, measuring, as true positive predictions, all of the classifications of the positive over the model's classification, indicating, respectively, the ability of the models to identify spam over the instances of spam.

The next relevant metric to be used is the F1-score, which represents the harmonic mean of precision and recall and gives a sustainable acknowledgment of the performance of the model, as it encompasses both types of metrics in a single number. This is extremely appropriate for imbalanced instances of a positive and negative type classification situation. Moreover, the confusion matrix will give a graphical representation of the models' performance depicting the true positive, true negative, false positive, and negative classification situations. This matrix will help in providing a detailed analysis of the classification decision situations of the different models and assist in pointing out the exact strong points and the points where the model still needs to be worked on to improve performance.

**4. Results and Analysis**

*4.1 Data Visualization*

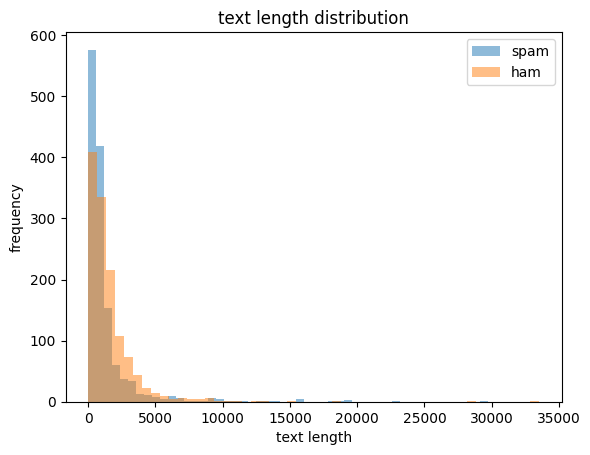
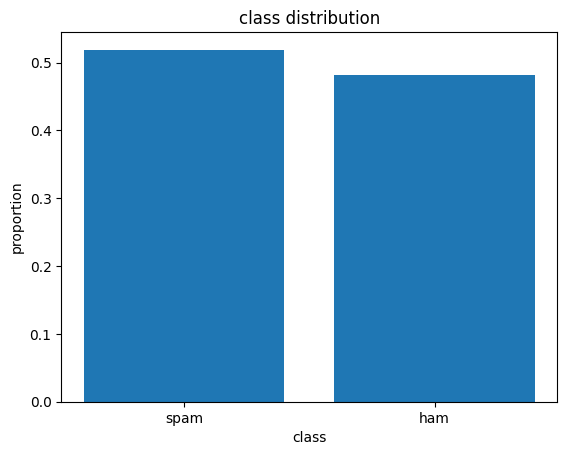


Fig. 2. Class Distribution Fig. 3. Text Length Distribution

The visualization on the dataset as seen in Fig. 2 displays us a noticeable distribution between spam and ham messages. The class distribution that is visible clearly shows that the spam instances are greater than the 0.5 threshold, whereas the ham messages are less than the previously mentioned threshold. Additionally, the text length distribution on Fig. 3 shows that the frequency of spam messages is relatively toward the smaller lengths and approaches close to 600 occurrences with lower frequencies on the longer text lengths. In contrast, the distribution of ham messages, which has a peak value of about 400 occurrences in smaller lengths, is a much lower frequency, thus signifying that ham messages are usually longer when compared to spam messages. The difference indicates that in the document length of a spam and ham message, the spam message usually being of shorter length.

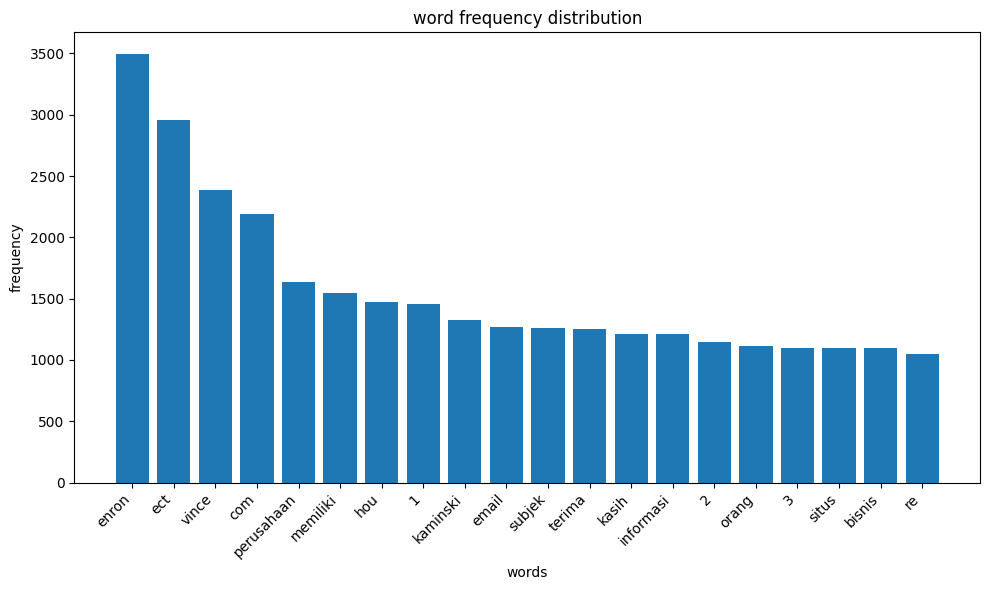


Fig. 4. Word Frequency Distribution

The chart above on Fig. 4 gives a great look at the word frequency distributions within the dataset. Hence, it provides good insights into the content. On top is the name "enron", which swamped the list with a whopping 3497 mentions, followed by "ect" at 2957 and "vince" at 2385. Notably, among such domain-specific lexicons, it is still possible to find universal linguistic elements, such as "com", or Indonesian words like "perusahaan" and "memiliki". Besides, the existence of numeric forms "1" and "2" and "3" somewhat suggests the application of quantitative numbers or references in the correspondence, giving a further dimension to the contents of the dataset. Besides, words such as "email" and "subjek" depict that it is their communicative nature that is mostly characterized as a dataset regarding its rich and diverse topics and discussions.

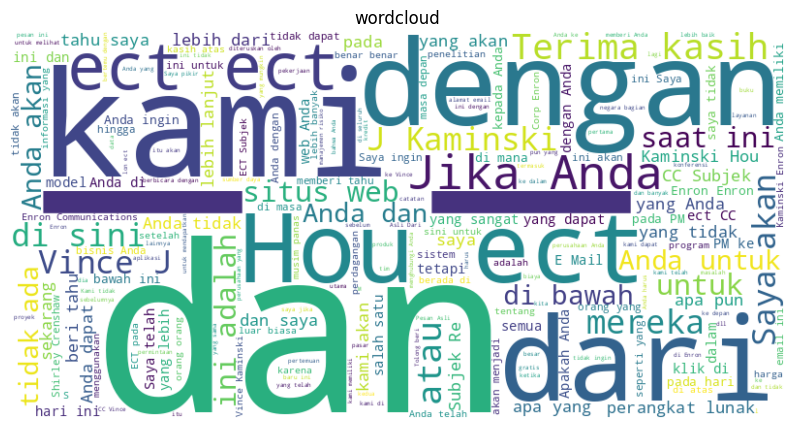


Fig. 5. Word Cloud Visualization

The word cloud visualization on the above figure on Fig. 5 displays the importance of each word which is in the dataset, meaning the bigger words have been used with higher frequency than smaller words. Here, many big words, such as "dan" and "kami", can be seen, so they are most likely the most common words. As another big words that can be seen that seem to be close in their size are "jika", "anda", "dari", and "dengan".

The n-grams lists are shown together alongside their frequency counts in the provided text. Looking at the most common through unigrams, using words or characters, it was shown to be that certain symbol, specifically "-", "\_", and "\*", can be seen quite frequently at 26432, 13788, and 2195, respectively. This makes them quite important in the text analysis, as they could be possibly indicating the break points or separators. Now, looking at all the bigrams, pairs of conjoined words, some combinations happening more than once, such as "Jika Anda" having an occurrence of 721 times, similarly "situs web" with 628 times and "di sini" having 617 occurrences.

**4.2 Results Comparison Analysis**

Table 1. Result for each model.

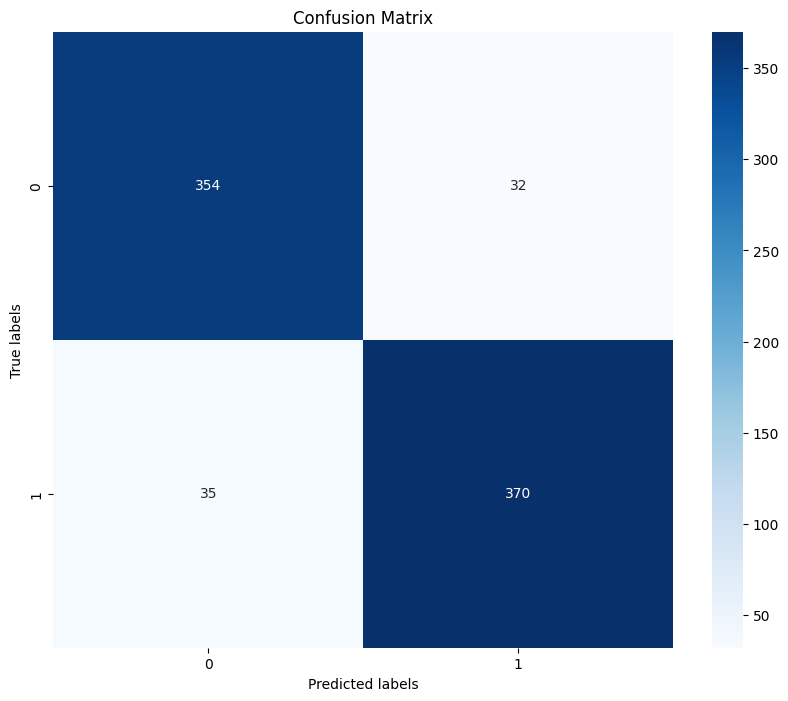
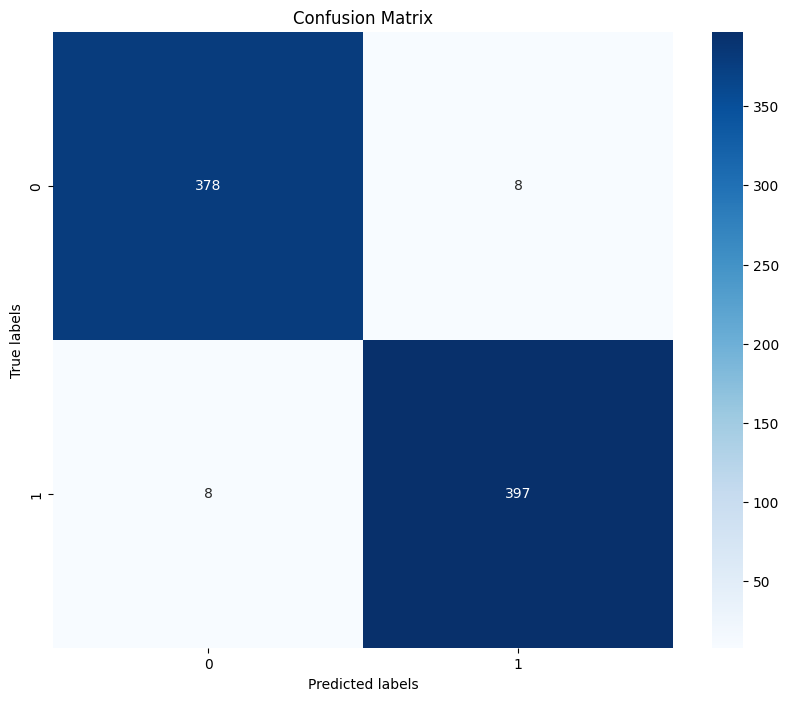
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Top-1 Accuracy | Accuracy | Precision | Recall | F1 Score |
| LSTM | Word2Vec | 0.91530 | 0.91534 | 0.91530 | 0.91530 |
| SVC Linear | Word2Vec | 0.89760 | 0.90019 | 0.89760 | 0.89732 |
| SVC RBF | Word2Vec | 0.91277 | 0.91485 | 0.91277 | 0.91257 |
| LSTM | TF-IDF | 0.97977 | 0.97977 | 0.97977 | 0.97977 |
| SVC Linear | TF-IDF | 0.97851 | 0.97858 | 0.97851 | 0.97850 |
| SVC RBF | TF-IDF | 0.97724 | 0.97735 | 0.97724 | 0.97724 |

The table 1. shows that the LSTM model, when executed with Word2Vec embeddings, comes up with remarkable accuracy of 91.53%, while precision has come up to 91.5%, recall is around 91.5%, and the F1 score is around 91.53%. The LSTM can capture the complex nuances of the text. It emerges as a good model in detecting spam emails. Similarly, Support Vector Machine (SVM) models using Word2Vec are found to give promising results, although a bit less than the LSTM. The linear SVC and SVC RBF models obtained accuracies of 89.76% and 91.28%, respectively, which were paralleled with their precision, recall, and F1- scores, near 90%. Although these were relatively less compared to the LSTM, the scores justify the efficacy of SVM in spam detection, where it is coupled with semantically rich embeddings like Word2Vec.

In fact, the results change significantly when using TF-IDF embeddings. LSTM initialized with TF-IDF embeddings surpasses other approaches and has a really notable accuracy of 97.97%, while precision, recall, and F1 scores are above 97.97%. Meaningful improvement shows the efficacy of the deep learning approach, especially when combined with embedding methodologies that give importance to the relevance of the word within the corpus, one among them being TF-IDF. Furthermore, the TF-IDF embeddings also produce promising results with SVC models. The SVC Linear model obtained with an accuracy of 97.85%, with precision, recall, and F1 scores around 97.86%. On the other hand, the SVM RBF model achieved an accuracy of 97.72%, and its precision, recall, and F1 scores around 97.74%. From these numbers, it is evident that the Support Vector Machine models perform quite robustly when paired with the TF-IDF embeddings and can deduce between spam and not-spam emails effectively. In addition, these results are a clear demonstration of the generalized learning of the TF-IDF embeddings in supporting accurate spam detection with different machine learning models.

As a sum of all the summaries, while Word2Vec embeddings seem to be promising, especially when paired with LSTM models, the large margin of performance improvement observed with TF-IDF embeddings underlines their importance in achieving optimal results on email spam detection tasks, especially under deep learning paradigms.

**4.3 Confusion Matrix Analysis**

****

(a) (b)

Fig. 6. (a) LSTM (TF-IDF), (b) LSTM (Wav2Vec)

Starting with the TF-IDF LSTM model on Fig. 6(a), it correctly identified 378 non-spam emails as non-spam, which shows its potential for correctly discerning non-spam content (true negatives). However, it also committed an error in classifying 8 non-spam emails as spam, that is, false positives, which means that this model slightly tends toward over classifying non-spam content into spam. On the front of spam classification, the TF-IDF LSTM model correctly detected 397 spam emails as spam (true positives), and thus it can be said that the model has attained a high level of proficiency while detecting spam content. However, it failed in detecting 8 spam emails as non-spam (false negatives), which indicates a minor deficiency that the model has in detecting certain instances of spam.

Now, focusing on the Word2Vec LSTM model on Fig. 6(b), similar patterns emerged, but the numbers were slightly different. This model got 354 non-spam emails correctly as non-spam (true negatives), which is slightly a lower accomplishment compared to the TF-IDF LSTM model. Additionally, it wrongly categorized 32 non-spam emails as spam, that is, false positives that demonstrates a slightly lower inclination towards misclassification in terms of that aspect. On the positive side, it correctly identified 370 spam emails as spam (true positives), which is in line with the performance in the TF-IDF LSTM model. However, it performed worse in terms of distinguishing spam from non-spam, as confirmed by its misclassification of 35 spam emails as non-spam (false negatives).

Comparing the two then, it is evident that they both have their areas of strength and a couple of weaknesses. The TF-IDF LSTM model performs better at distinguishing non-spam content, as seen from the high number of true negatives. In contrast, Word2Vec LSTM has a high false positive rate for non-spam emails. The two are closer in marking spam content, though they do differ slightly in the false negative rate.

**5. Conclusion**

The experiment that we conducted on the Indonesian email dataset, to categorize the email based on spam or not, has revealed an interesting insight. The result of the experiment that compared two different word embeddings revealed that TF-IDF is superior to Word2Vec in terms of accuracy. This gap in difference between the two-word embeddings model isn't as close as we think, because the gap is almost 6%, which has been slightly minimized because of the hyperparameter tuning in the Word2Vec model. Furthermore, by utilizing the power of the models and feature extraction in a very detail-oriented way made possible by the use of Term Frequency-Inverse Document Frequency, the LSTM model obtained a remarkable accuracy of 97.98%, very close to that of the SVM model at an accuracy difference of just 0.2%.

For next research, this can be boosted by adding several advanced natural language techniques, such as using contextual embeddings from BERT or GPT. The next investigation could be on the impact of various methods for feature selection and the trial of various neural network architectures. If implemented live in email environments, this should be the most practical testing to assure the actual effectiveness and efficiency of the developed models and on the best ways to implement them on a large scale for the interoperability and augmentation of cybersecurity.

**References**

Abdulrahman, S. H., & Salim, M. (2022). Using Decision Tree Algorithms in Detecting Spam Emails Written in Malay: A Comparison Study. *ITM Web of Conferences*, *42*. https://doi.org/10.1051/itmconf/20224201001

Adnan, M., Imam, M. O., Javed, M. F., & Murtza, I. (2024). Improving spam email classification accuracy using ensemble techniques: a stacking approach. *International Journal of Information Security*, *23*(1). https://doi.org/10.1007/s10207-023-00756-1

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). Experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*. https://doi.org/10.1145/345508.345569

Anugroho, P., Winarno, I., & Rosyid, M. N. (2005). Klasifikasi email spam dengan metode naïve bayes classifier menggunakan java programming. *Journal Informatika*.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3). https://doi.org/10.1023/A:1022627411411

Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, *5*(6). https://doi.org/10.1016/j.heliyon.2019.e01802

Hidayat Informatika, A. (2023). Klasifikasi Spam Email Menggunakan Metode Naive Bayes. *Teknologipintar.Org*, *3*(2).

Hochreiter, S., & Schmidhuber, J. (1997). Long Short Term Memory. Neural Computation. *Neural Computation*, *9*(8).

Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology (Singapore)*, *11*(2). https://doi.org/10.1007/s41870-018-0157-5

John-Africa, E., & Emmah, V. T. (2022). Performance Evaluation of LSTM and RNN Models in the Detection of Email Spam Messages. *European Journal of Information Technologies and Computer Science*, *2*(6). https://doi.org/10.24018/compute.2022.2.6.80

Pamungkas, A. C. (n.d.). *Spamming in Indonesian Criminal Law Perspective: The Nature Elements*. *7*(2), 225–238. https://doi.org/10.15294/lrrq.v7i1.43167

Pedersen, E. O., & Brincker, M. (2021). Philosophy and Digitization: Dangers and Possibilities in the New Digital Worlds. *SATS*, *22*(1). https://doi.org/10.1515/sats-2021-0006

Rahman, S. E., & Ullah, S. (2020). Email Spam Detection using Bidirectional Long Short Term Memory with Convolutional Neural Network. *2020 IEEE Region 10 Symposium, TENSYMP 2020*. https://doi.org/10.1109/TENSYMP50017.2020.9230769

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *Learning for Text Categorization: Papers from the AAAI Workshop*, *WS-98-05*(Cohen).

Shaik, C. M., Penumaka, N. M., Abbireddy, S. K., Kumar, V., & Aravinth, S. S. (2023). Bi-LSTM and Conventional Classifiers for Email Spam Filtering. *Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2023*. https://doi.org/10.1109/ICAIS56108.2023.10073776

Zhang, L. E., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, *3*(4). https://doi.org/10.1145/1039621.1039625