khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Android-Based Short Message Service Filtering using Long Short-Term Memory Classification Model

M. Laylul Mustagfirin [1], Giri Wahyu Wiriasto [2*], I Made Budi Suksmadana [3], Indira Puteri Kinasih [4]

[1,2,3]Departement of Electrical Engineering
University of Mataram
Mataram
[4]Departement of Mathematics Education
Universitas Islam Negeri Mataram
Mataram
*giriwahyuwiriasto@unram.ac.id

Abstract-Short Message Service (SMS) is a technology for sending messages in text format between two mobile phones that support such a facility. Despite the emergence of many mobile text messaging applications, SMS still finds its use in communication among people and broadcasting messages by governments and mobile providers. SMS users often receive messages from parties, particularly for marketing and business purposes, advertisements, or elements of fraud. Many of those messages are irrelevant and fraudulent spam. This research aims at developing android-based applications that enable the filtering of SMS in Bahasa Indonesia. We investigate 1469 SMS text data and classify them into three categories: Normal, Fraudulent, and Advertisement. The classification or filtering method is the long short-term memory (LSTM) model from TensorFlow. The LSTM model is suitable because it has cell states in the architecture that are useful for storing previous information. The feature is applicable for use on sequential data such as SMS texts because every word in the texts constructs a sequential form to complete a sentence. The observation results show that the classification accuracy level is 95%. This model is then integrated into an Android-based mobile application to execute a real-time classification.

Keywords: text filtering, recurrent neural network, long short term memory

Article info: *submitted April 4, 2022, revised June 11, 2022, accepted September 23, 2022*

1. Introduction

Short message service (SMS) is a communication service in text format that has been used by humans in the last few decades and has become an embedded feature on every cellphone, be it a featured phone or smartphone. Since it is a service that has advantages such as low cost and eases to use, this service is also used by certain parties to send an unwanted text message, namely, spam message [1,2]. Spam is a type of message that is sent arbitrarily with various purposes such as promotions/advertising, borrowing money, announcements of sweepstakes, and such so that they are disturbing to mobile phone users [3], [4]. Spam message itself has been found in many countries including Indonesia. In 2019, Indonesia was included in the top 20 countries with the highest number of spam text messages in the world with an average mobile phone user in Indonesia receiving 46 spam messages every month [5].

There are several ways that have been done by the government, operators, and researchers to overcome spam message attacks. It can be prevented by means of operators filtering all text messages sent through SMSC (short message service center) or by installing a system that can detect spam text when it is sent to end-users [2]. Handling with the system installed at the end-user can then be done by filtering based on the content of the message received [6]. This method involves a classification method that is part of machine learning applications such as those that have been applied to spam e-mail filtering [3].

Machine learning methods have been used in several previous studies to classify spam text in Indonesian. Setifani et al. [7] compared the naive Bayes algorithm, SVM and decision tree in classifying text messages into three different classes. Herwanto et al. [8] classified Indonesian spam text messages using the multinomial naive Bayes algorithm and produced an F1-score of 0.93. The multiclass classification of the text message was carried out by Theodorus et al. [9] by comparing the performance of several different models with an average accuracy of 94% obtained. The research was also conducted using a deep learning architecture conducted by Tandra et al [10] which compared the Multinomial Naïve Bayes capability with the Bi-Directional LSTM algorithm. All of these studies were carried out up to testing how the model's performance in classifying text messages according to their type.

The focus of this paper is on implementing the trained model to classify Indonesian text or '*Bahasa*' messages into three categories that are listed as the most popular text message type in Indonesia. It is in line with the study proposed by Sethi and Bhootna [11] regarding spam text filtering applications on android devices using Bayesian algorithms. Uysal et al. [12] did the same thing by conducting research on spam and non-spam text message filtering applications using the Bayesian method. Then there is also a study to create a mobile-based system for filtering spam text in English – India by Yadav et al using the SVM method [13].

Based on the success of previous studies in testing various methods to overcome spam text, we propose to make a model using the Long short-term memory (LSTM) method [14] and implement it in the form of an Android-based application. The LSTM method is a model designed to handle sequential data that depends on the ordered data such as word sequences with various lengths and is able to capture long-term dependencies of sentences on the data [15]. In this study, the authors classify Indonesian-language *('bahasa')* short messages service into three classes, namely normal/personal text, fraudulent text, and promotional text. The division into three classes is because the SMS received by cellphone users in Indonesia are received from known people (personal SMS), sent by unknown parties with the aim of deceiving the recipient, and SMS containing promotions or advertisements from third parties in the form of companies [7,8]. This research was conducted to see how the performance of the model when put together into a complete system that can classify incoming SMS in real-time and not just to test how effective and accurate the model is. With this research, it is hoped that the resulting system can be a reference for smartphone users. in managing incoming SMS so that no more users are exposed to fraud.

This paper consists of four parts. The first part explains the background and initial idea of this research as a developmental idea from the previous studies. The second part is the part that explains how the research is carried out or the research methods used. The third section describes the results of the research and a discussion based on the research results. Finally, the fourth section contains the conclusions obtained from the research that has been carried out.

2. Methods

In the previous studies, many attempts to classify short text messages using artificial intelligence to deal with spam have been carried out. In this research, the development of a system that has been combined with an artificial intelligence model and installed on the end-user's device is carried out. Figure 1 shows a general illustration of how this mobile application works where the authors in this study focus on the part that is inside the box with the dotted black line.

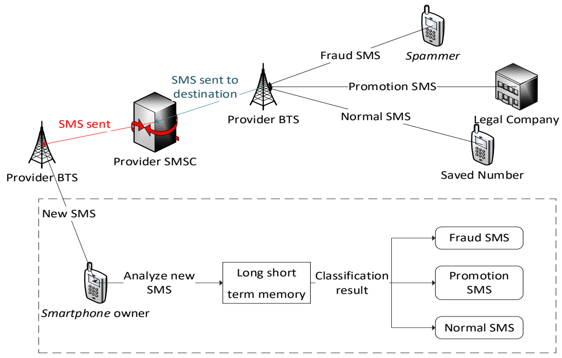


Figure 1. Implementation System

a. Data Collection

The text dataset used in this study was obtained in two ways, namely collection with the help of respondents who filled out a questionnaire form using the google form media and the second was obtained from the Kaggle website. The total data obtained are 1469 sentences of data text. The labeling of the data obtained from the questionnaire was carried out by the researcher based on the closeness of the meaning of the Indonesian text messages. The dataset is grouped into three classes, namely normal, fraudulent, and advertisement text messages with a total of 1469 data. The data is divided into training data (75%) and test data (25%). The division of each class in the dataset can be seen in Table 1.

Table 1. Text Message amount per category

Category	amount
Normal	537
Fraud	334
Advertisement	598
Total	1469

In table 2 we present 15 unprocessed data which are divided into three different classes.

Table 2. Example of Text Message Data Sets

Text Message (in bahasa)	Translated Mesage
NASABAH Yth! Anda diminta ke Kantor CABANG untuk monitor ulang REKENING TABUNGAN yg ERROR SISTEM. Sebelumnya harap telp dl pak WAWAN 082317744737	Dear CUSTOMERS! You are asked to go to the BRANCH office to re-monitor the SAVING ACCOUNT because SYSTEM ERROR. Before that, please call Mr. WAWAN 082317744737

Text Message (in bahasa)	Translated Mesage
INFO RESMI PT.TRI CARE PIN Pemenang (8jf2177) anda m-dptkan HADIAH I unit TOYOTA YARIS U/Info cek pin klik: *www.tricare2015.blogspot.com*	OFFICIAL INFO PT. TRI CARE PIN Winner (8jf2177) you get GIFT I unit TOYOTA YARIS U/Info check pin click: *www.tricare2015.blogspot.com*
Anda M'dptkan $ubs!d! Dri Pert4min4 Rp.189 jt Pin (717747) !nfo Wh4ts4pp:085243235227 atau Surat Keputusan dari Tri Care Indonesia No.XV/2015 Pin_Pemenang ANDA : 67ytg44 mendapatkan hadiah cek tunai Rp 45 jt. U/INFO kunjungi: *www.id.tri.webnode.com.*	You get allowance from Pertamina Rp. 189 million Pin (717747) Info Whatsapp: 085243235227 or Decision Letter from Tri Care Indonesia No.XV/2015 YOUR Winner_Pin : 67ytg44 get cash reward Rp 45 million U/INFO visit: *www.id.tri.webnode.com.*
ASS..YTH BPK/IBU BTUH BIAYA TMBHAN UNTUK MDAL USAHA DLL U/INFO/SILAHKAN CHAT WA : 085387120337	ASS.. DEAR MR/MRS NEED ADDITIONAL COSTS FOR BUSSINESS CAPITAL ETC U/INFO/PLEASE CHAT WA : 085387120337
Yuk Ikuti akun dakwah, caranya: ketik IKUT [spasi] AkuCintaIslam kirim ke 082110001021.	Let's Follow da'wah account, step: type IKUT [space] AkuCintaIslam send to 082110001021.
Bebas Pulsa! Ambil bonusmu di *600# (GRATIS). Dptkan gratis nelpon atau internetas atau promo lainnya sesuai hobimu!	Free of charge! Take your bonus at *600# (FREE). Get free calls or internet or other promos according to your hoby!
Ayo klik tsel.me/maxendeal30gb utk kuota MAXstream 30GB hny 40rb dan tonton FA CUP: Leicester City VS MU atau Liga Serie A: Roma VS Napoli di MAXstream	Come click tsel.me/maxendeal30gb for quota MAXstream 30GB just rp. 40k dan watch FA CUP: Leicester City VS MU or Serie A league: Roma VS Napoli on MAXstream
Bebas online seharian dgn Unlimited Internet hanya Rp105rb/bln! Daftar XL PRIORITAS dgn chat ke wa.me/62818800055/?tex-t=gabung. S&K berlaku.	Free online all day with Unlimited Internet only Rp 105k/month! Register XL PRIORITAS by chat to wa.me/62818800055/?text=join. T&C apply.
Yuk tetap gunakan Flash Volume Ultima utk update informasi Anda, kuota 60MB/7hr mulai Rp7rb di *100*431#. Tarif&lokasi cek di tsel.me/FL	Let's keep use Flash Volume Ultima to update your information, quota 60MB/7 day start from Rp 7k at *100*431#. Check rates & locations at tsel.me/FL
Nikmati nelpon dan SMS UNLIMITED ke sesama Indosat Ooredoo internetan sepuasnya 3 hari dengan paket Ramadhan Unlimited. ketik *123*88#	Enjoy UNLIMITED calls and SMS to other Indosat Ooredoo fellow internet all you want for 3 days with the Ramadhan Unlimited package. Type *123*88#

Text Message (in bahasa)	Translated Mesage
Aku senin udah ke tempat kerja. Minggu2 depan aku gaktau bisa/ngga :(	I'm at work Monday. I don't know I can or can't next week
Maaf kaprodi kita itu siapa yah hehe	Sorry, who is the head of our study program?
Pada ga aktif si tombol algoritma2 nya teh ga ngerti	The algorithm buttons not active, I don't understand
Gara gara batman. Pdhl udh diulang	Because of batman. Even though it's been repeated
Mohon maaf bang, kemarin untuk kelompok saya diberitahu kirim lewat chat, kami juga ndak tau alamat email abang	Sorry, yesteday my group was told to send via chat, we also don't know your email address

b. Data Classification

The data that has been obtained is then classified into three parts, namely training data, data validation and data testing. Data distribution was carried out using a training data ratio of 60%, data validation of 15%, and test data of 25% where data validation was used after each epoch was completed to see whether the model was overfitting or not.

c. Pre-Processing

In the pre-processing stage, the stages are carried out to generalize the format of the text. Pre-processing of text message sentences is conducted through four stages, namely punctuation removal, case folding, stopwords removal and tokenization. Then the text data will be converted into a number representation. Table 3 shows the data that has gone through the pre-processing stage of each class.

Table 3. Pre-processing Stage Data Example

SMS Text (in bahasa)	Class
Ass yth bpk ibu btuh biaya tmbhan untuk mdal usaha dll u info silahkan chat wa 085387120337	fraud
bebas pulsa ambil bonusmu di 600 gratis dptkan gratis nelpon atau internetas atau promo lainnya sesuai hobimu	advertisement
aku senin udah ke tempat kerja minggu2 depan aku gaktau bisa ngga	Normal

d. Word Embedding

Word embedding is a method used to represent words in a vector form consisting of real numbers. The vectors can then be plotted to see where the words are and words that have similarities will have a close position when plotted. Word embedding itself has a network similar to an ordinary neural network and is often used as input in deep learning models for solving NLP problems [3]. There are also word embedding models that have been trained previously and can be directly used, such as the Glove model [16] and the Word2Vec model [15]. Figure 2 shows examples of words that are plotted in a two-dimensional graph after their representation is generated using word embedding. It can be seen that the word "cat" has a close distance from the word "dog" because it has a relationship that is both including animals. While the word "computer" has a long distance from the word "tomatoes" because it has a much different contextual nature, namely "computer" is a tool while "tomatoes" is a vegetable. In this study, the author chose to train the word embedding model himself because the dataset used was in Bahasa.

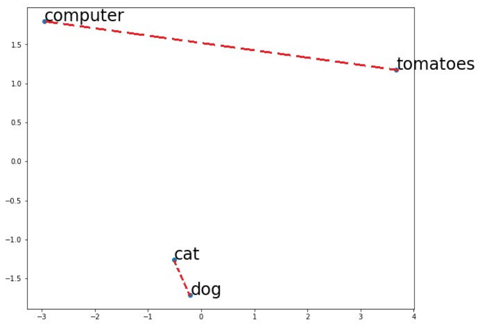


Figure 2. Words Plotting as the result of word embedding

In this study, the maximum number of words contained in each data is 17 words. The embedding layer used has an output dimension of 17x128. The output matrix of the embedding layer has 17 rows where each row represents the 17 words. Figure 3 shows an example of the embedding layer output.

1st word → [-2.72e-02 -2.46e-02 -2.285e-02 3.298e-02 -3.102e-02 3.677e-03 … -3.9e-02]
2nd word → [2.124e-02 -2.736e-02 4.604e-02 9.968e-03 1.386e-02 1.615e-02 … -2.4e-02]
3rd word → [2.519e-02 -3.179e-02 2.636e-02 -1.705e-02 -2.655e-02 3.463e-04 … 1.2e-02]
⋮ [⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮]
15th word →[2.210e-02 1.464e-02 1.673e-02 -4.484e-02 3.380e-02 4.852e-02 … 1.92e-02]
16th word →[-4.54e-02 3.226e-02 1.951e-02 -9.576e-03 -3.482e-02 3.880e-02 … 1.06e-02]
17th word →[1.00e-02 3.647e-02 8.621e-03 -4.725e-02 -4.777e-02 -3.587e-04 … 2.77e-02]

Figure 3. The output matrix of the embedding layer

e. Model Implementation

The model implementation process is carried out using the Tensorflow library provided by Google Collaboratory [17]. Google Collaboratory is a tool built on Jupyter Notebook. Jupyter Notebook is a tool that runs on a browser and has integration in interpreted languages such as python complete with libraries for data processing [18]. Google collaboratory or google collab is a product of the Google team with the aim of simplifying work related to machine learning, data analysis, and education based on the Jupyter Notebook [19,20]. Figure 4 shows the network architecture used in this study.

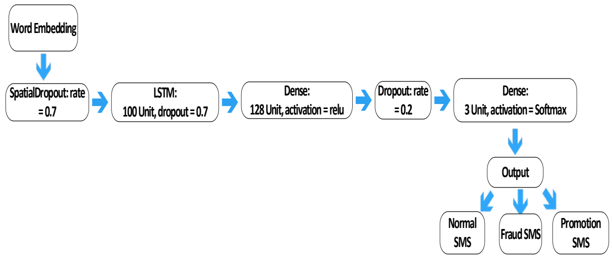


Figure 4. Proposed Model Design

f. Dropout and Spatial Dropout

Dropout is one way to regularize the model so as not to overfit by not randomly involving nodes in the training process so that they do not depend on each other [21]. However, in certain cases, a variant of dropout is used, namely spatial dropout. Spatial dropout is a dropout method that does not include all feature mappings compared to doing a dropout on randomly selected nodes. This is done because the activation of feature mapping has a strong correlation such as in image data or text data so that the ordinary dropout method does not have much effect [22].

g. Long Short-Term Memory

Long short-term memory is a modified version of the Recurrent Neural Network (RNN) model. The RNN architecture itself has a long-term dependency problem that causes the model to be unable to process sequential data that is too long/has a large time step difference [23]. The LSTM network consists of repeating units where each unit consists of parts called gates that determine the addition or subtraction of information from the data, consisting of input gate (), forget gate () and output gate () as shown on figure 5 and figure 6. The following is a transition function that exists in the LSTM sections [24]:

$$i_t = \sigma(W_i \cdot [h_{t-1} + b_i])$$
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$
$$q_t = \tanh\left(W_q \cdot [h_{t-1}, x_t] + b_q\right)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot q_t$$
$$h_t = o_t \odot tanh(c_{t)}$$

Where is a sigmoid function, and is the weight and bias of each gate which will continue to be updated during the training process, is a vector in the cell state section, is a hidden state in the previous unit and is an operator for element-wise multiplication [1,25].

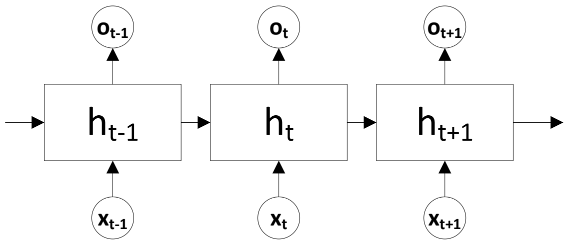


Figure 5. LSTM Network Structure

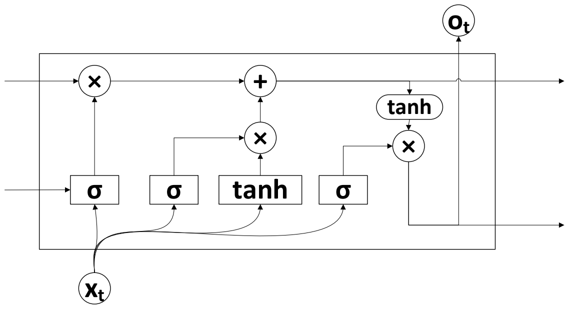


Figure 6. LSTM single unit architecture on the network

h. Dense

Dense is a layer that connects all neurons directly to each other with the next layer and can often also be referred to as a fully connected layer. In the model proposed by the authors, a dense layer is used in the final two layers, a hidden layer with 128 neurons and a Rectified Linear Unit (ReLU) activation function. The ReLu function returns if it is positive and 0 otherwise [26].

$$f(x) = max\,(0, x) \tag{7}$$

The reason for using a dense layer as a hidden layer is because it can improve the performance of the model in classification [27]. Dense layer is also used as the last layer or layer that gives the output of model predictions. Because in this study a multiclass classification was used, three neurons were used and the softmax activation function was employed [28].

$$softmax(x) = \frac{\exp\,(x_i)}{\sum_{j=1}^{n} \exp\,(x_j)} \tag{8}$$

i. Model Evaluation

To determine the performance of the model, standard metrics such as confusion matrix, accuracy, precision, recall, false-positive rate, F1-score, Receiver Operating Characteristics (ROC) Curve and Area Under The Curve (AUC) are used. These size indications are based on the method designed by Kohavi and Provost [29]. Then from the confusion matrix, the other metrics mentioned above can be calculated [30].

j. Designing Phase

As part of the implementation phase, the author develops a smartphone application for devices with Android OS. This application development was carried out using Android Studio with the Kotlin programming language on a Personal computer running Windows 10 with an Intel Core 13-9100f processor, 8 GB of RAM and an NVIDIA 1050TI graphics card. For testing during the development process, an emulator device provided directly by Android Studio was used, namely the Google Pixel 4 XL and the OPPO A92 physical device.

After the LSTM model has been trained and tested until it reaches the desired results, the model is exported using the TensorFlow Lite library. TensorFlow Lite is a tool developed by the TensorFlow team so that machine learning models can run on devices that have specifications below ordinary computers/PCs such as mobile, embedded and IoT devices [31]. The exported model is then combined with the application so that when a new SMS comes in, it will be preprocessed then the model will predict the class of the new SMS and finally group it together with other SMS with the same class. The workings of the system are illustrated using a flow chart which can be seen in figure 7.

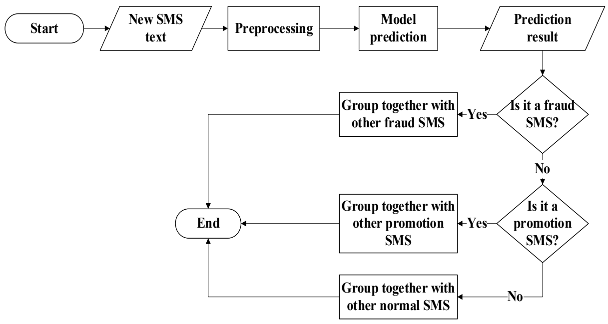


Figure 7. Flowchart of how the app works

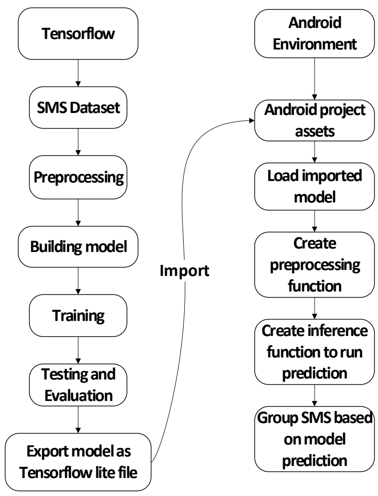


Figure 8. The process of integrating the model into the Android system

Figure 8 shows the steps involved when a model is added to an Android application. After the dataset is obtained, preprocessing is carried out, then the model is built according to the proposed initial design. The model is then trained, tested and evaluated until a satisfactory performance is found. To make the model usable on other devices such as Android, TensorFlow provides a library to convert the trained model into TensorFlow lite format. The TensorFlow lite file is imported into the android project directory.

In the Android application design process, the next step is to create a preprocessing function that is the same as the preprocessing function used when the model is trained to have an accurate output. The incoming SMS text then goes through the preprocessing stage and the model will make predictions on the data. The prediction results of the model determine which category the SMS falls into.

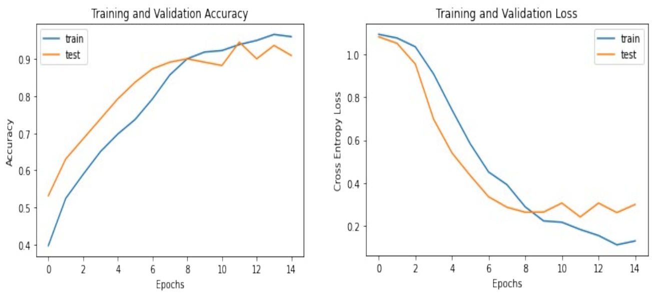
3. Results

a. Model Evaluation

The model evaluation process is carried out to see how well the model performs after going through the training process. Figure 6 shows a graph of the comparison of accuracy and loss to the increase in the number of epochs on the training data and validation data.

It can be seen in figure 9(a) that there is an increase in the value of accuracy as the number of epochs increases and in figure 9(b) there is a decrease in the value of loss/ error when the number of epochs increases so that a convergent graph is obtained. Then testing using 20% of the data that has never been seen by the previous model. The results of model testing in the form of a confusion matrix can be seen in figure 10.

Based on the confusion matrix, table 4 shows the results of the metrics used to test the model's performance. The model obtains an accuracy level of 0.951, a precision value of 0.936, a recall value or true positive rate of 0.946, a false positive rate of 0.026, an F1-score as the average between precision and recall of 0.941 and the last ROC-AUC value of 0.959. In figure 11 we present the ROC curve of the LSTM model that has been trained.



(a) (b)

Figure 9. Model Trained Result

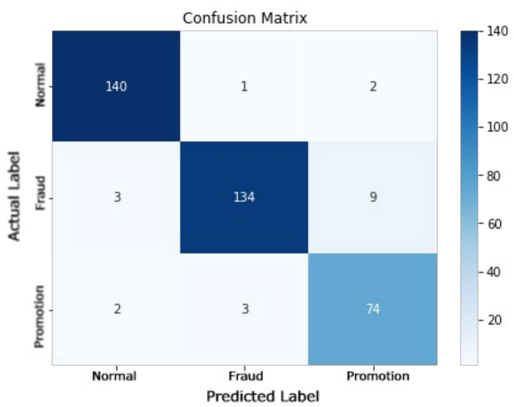


Figure 10. Confusion matrix of tested model

Table 4. Metrics Value of Model Evaluation Result

Metrik	Result
Accuracy	0.951
Precision	0.936
Recall/TPR	0.946
FPR	0.026
F1-Score	0.941
ROC-AUC	0.959

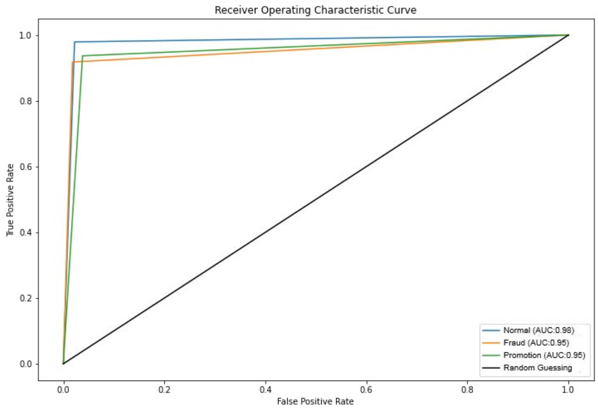


Figure 11. ROC-AUC model

Then in table 5 we compare the results of the LSTM model that we have trained with other machine learning (ML) models that have been carried out by Herwanto et al [8] and Setifani et al [7] using the same dataset. The LSTM model appears to have the highest value for each of the test metrics used in all models.

Table 5. Comparison of results from several classification models

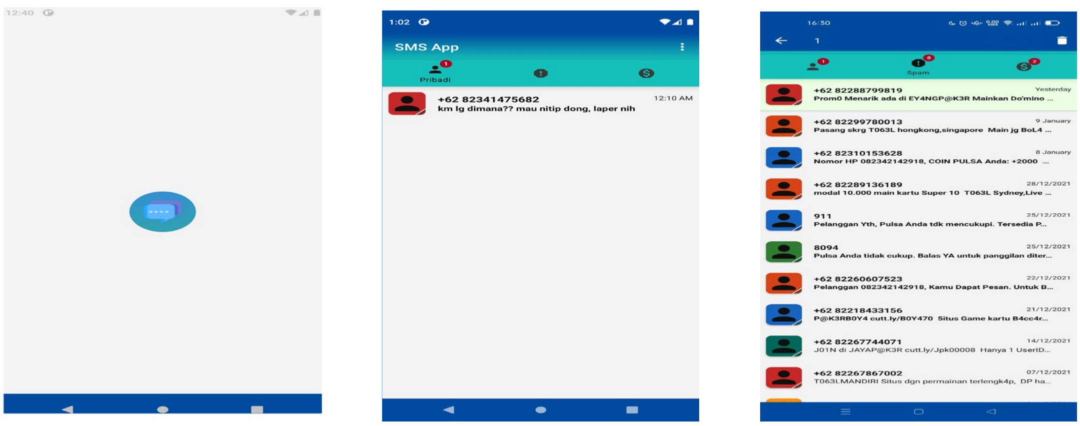
Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes[8]	0.94	0.92	0.93	0.93
Decision Tree[8]	0.87	0.87	0.84	0.85
SVM[8]	0.93	0.93	0.92	0.92
MNB[7]	0.94	0.93	0.92	0.93
Random Forest[7]	0.93	0.92	0.93	0.92
LSTM	**0.95**	**0.93**	**0.94**	**0.94**

Based on table 5. it can be seen that the LSTM model proposed in this study has the best performance compared to other models. The highest accuracy is 0.95 obtained by the LSTM model, followed by Naïve Bayes, SVM, Random Forest and Decision Tree. For precision metrics, the LSTM model obtained the same value with 0.93 for the SVM and MNB models, followed by Naïve Bayes, Random Forest and Decision Tree. Then for recall, the LSTM model has the highest value obtained 0.94 followed by Naïve Bayes, Random Forest, SVM, MNB and Decision Tree. Meanwhile, the F1-score of the LSTM model also obtained the highest score of 0.94, followed by Naïve Bayes, MNB, SVM, and finally Decision Tree.
Based on the test results, it can be seen that the LSTM model has the best performance compared to other models.

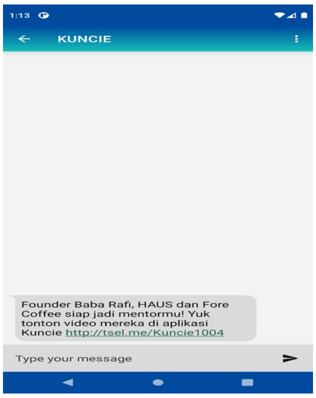
b. Mobile Application Interface

This application is a direct implementation of the LSTM model that has been trained so that it can classify Indonesian-language text messages directly when a new text is received. The main part of the application is divided into three parts, namely a page that displays a list of normal, fraudulent, and also advertising text message. Navigating between pages can be done by swiping

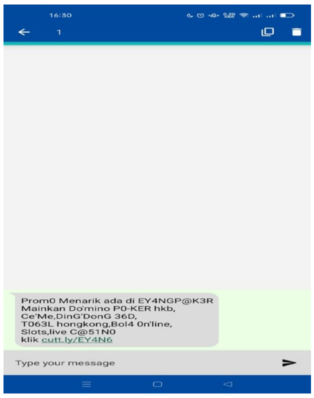
the screen to the left or right. To view the message content, you can do this by tapping the desired message so that the message opens and the view will change to a page containing the conversation between the recipient and the sender of the message. The application screen display can be seen in figure 12.



(a) (b) (c)



(d)



(e)

Figure 12. Design of android Apps Interface

Figure 12(a) shows the appearance of the application icon on the smartphone screen when the application is first opened. Figure 12(b) is a classified text message list display screen with the initial display screen on the normal text message list page. Figure 12(c) is an additional menu display screen when a message is selected, the operation that can be performed is to delete the selected message. Figure 12 (d) is a screen that displays the contents of the message that is opened and contains a column to type and a button to send a message. Figure 12(e) is an additional menu display screen when an item in a conversation is selected, the operations that can be performed are to copy or delete the selected item.

4. Conclusion

In this study, the authors propose an LSTM model combined with an Android-based mobile application to filter Indonesian text messages according to their type. This research produces a model that performs well and applies to an android application. The evaluation suggests that the level of accuracy possessed by the LSTM model when trained using the Indonesian text message dataset was 0.951. The results obtained in previous studies using other methods have shown high accuracy. However, the studies limited themselves to examining the effectiveness of methods without implementation as an application. The application can directly classify incoming text messages in real time and group them into the specified message list.

References

[1] X. Liu, H. Lu, and A. Nayak, “A Spam Transformer Model for SMS Spam Detection,” *IEEE Access*, vol. 9, pp. 80253–80263, 2021, doi: 10.1109/ACCESS.2021.3081479.

[2] M. T. Nuruzzaman, C. Lee, and D. Choi, “Independent and personal SMS spam filtering,” in *Proceedings - 11th IEEE International Conference on Computer and Information Technology, CIT 2011*, 2011, pp. 429–435. doi: 10.1109/CIT.2011.23.

[3] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, “A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages,” *Future Internet*, vol. 12, no. 9, Sep. 2020, doi: 10.3390/FI12090156.

[4] C. Khemapatapan, “Thai-English spam SMS filtering,” in *2010 16th Asia-Pacific Conference on Communications, APCC 2010*, 2010, pp. 226–230. doi: 10.1109/APCC.2010.5679770.

[5] “Truecaller Insights: Top 20 Countries Affected by Spam Calls & SMS in 2019 - Truecaller Blog.” https://truecaller.blog/2019/12/03/truecaller-insights-top-20-countries-affected-by-spam-calls-sms-in-2019/ (accessed Jun. 14, 2022).

[6] J. M. G. Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, “Content based SMS spam filtering,” in *Proceedings of the 2006 ACM Symposium on Document Engineering, DocEng 2006*, 2006, vol. 2006, pp. 107–114. doi: 10.1145/1166160.1166191.

[7] N. A. Setifani *et al.*, “PERBANDINGAN ALGORITMA NAÏVE BAYES, SVM, DAN DECISION TREE UNTUK KLASIFIKASI SMS SPAM,” 2020.

[8] H. Herwanto, N. L. Chusna, and M. S. Arif, “Klasifikasi SMS Spam Berbahasa Indonesia Menggunakan Algoritma Multinomial Naïve Bayes,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 4, p. 1316, Oct. 2021, doi: 10.30865/mib.v5i4.3119.

[9] A. Theodorus, T. K. Prasetyo, R. Hartono, and D. Suhartono, “Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia,” in *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, Apr. 2021, pp. 199–202. doi: 10.1109/EIConCIT50028.2021.9431859.

[10] V. G. Tandra, Y. Yowen, R. Tanjaya, W. L. Santoso, and N. Nurul Qomariyah, “Short Message Service Filtering with Natural Language Processing in

Indonesian Language," Aug. 2021. doi: 10.1109/ICISS53185.2021.9532503.

[11] G. Sethi and V. Bhootna, "SMS Spam Filtering Application Using Android." [Online]. Available: www.ijcsit.com

[12] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "A novel framework for SMS spam filtering," 2012. doi: 10.1109/INISTA.2012.6246947.

[13] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta, and V. Naik, "SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering," in *HotMobile 2011: The 12th Workshop on Mobile Computing Systems and Applications*, 2011, pp. 1–6. doi: 10.1145/2184489.2184491.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. 'A Ranzato, "Learning longer memory in recurrent neural networks," 2015.

[16] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1532–1543. doi: 10.3115/v1/d14-1162.

[17] USENIX Association., ACM SIGMOBILE., ACM Special Interest Group in Operating Systems., and ACM Digital Library., *Papers presented at the Workshop on Wireless Traffic Measurements and Modeling : June 5, 2005, Seattle, WA, USA*. USENIX Association, 2005.

[18] B. M. Randles, I. v. Pasquetto, M. S. Golshan, and C. L. Borgman, "Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study," Jul. 2017. doi: 10.1109/JCDL.2017.7991618.

[19] "Google Colab." https://research.google.com/colaboratory/faq.html (accessed Jun. 14, 2022).

[20] T. Carneiro, R. V. M. da Nobrega, T. Nepomuceno, G. bin Bian, V. H. C. de Albuquerque, and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Jul. 2012, [Online]. Available: http://arxiv.org/abs/1207.0580

[22] J. Tompson, R. Goroshin, A. Jain, Y. Lecun, and C. Bregler, "Efficient Object Localization Using Convolutional Networks."

[23] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," May 2015, [Online]. Available: http://arxiv.org/abs/1506.00019

[24] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM Neural Network for Text Classification," Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08630

[25] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, 2013, pp. 273–278. doi: 10.1109/ASRU.2013.6707742.

[26] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines."

[27] V. L. Helen Josephine, A. P. Nirmala, and V. L. Alluri, "Impact of Hidden Dense Layers in Convolutional Neural Network to enhance Performance of Classification Model," *IOP Conference Series: Materials Science and Engineering*, vol. 1131, no. 1, p. 012007, Apr. 2021, doi: 10.1088/1757-899x/1131/1/012007.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*.

[29] R. Kohavi, "Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process," 1998. [Online]. Available: http://robotics.stanford.edu/~ronnyk/glossary.html

[30] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340–341, pp. 250–261, May 2016, doi: 10.1016/j.ins.2016.01.033.

[31] "TensorFlow Lite." https://www.tensorflow.org/lite/guide (accessed Jun. 14, 2022).