The Comparison of Deep Learning Models for Indonesian Political Hoax News Detection

Oktavia Citra Resmi Rachmawati[1]* and Zakha Maisat Eka Darmawan[2]
[1]Department of Information and Computer Engineering,
The Electronic Engineering Polytechnic Institute of Surabaya
Surabaya, Indonesia 60111
[2]Department of Creative Multimedia Technology,
The Electronic Engineering Polytechnic Institute of Surabaya
Surabaya, Indonesia 60111
Email: [1]citra.research@gmail.com, [2]zakha@pens.ac.id

Abstract—**Indonesia is the world's fourth most populous country and has a diverse sociopolitical landscape. Political fake news exacerbates existing social divisions and causes political polarization in Indonesian society. Hence, studying it as a specific challenge can contribute to broader discussions on the impact of fake news in different contexts. The researchers propose a hoax news detection system by developing a deep learning model with various lapses against a data set preprocessed using term-frequency and token filtering to represent the most prominent words in each class. The researchers compare the layers with the potential to have high performance in predicting the falsity of Indonesian political news data by observing the models based on training history plots, model specification, and performance metrics in the classification report module. The deep learning models include One-Dimensional Convolution Neural Networks (1D CNN), Long-Term Short Memory (LSTM), and Gated Recurrent Unit (GRU). The news data are obtained from the Kaggle site, containing 41.726 rows of data. Based on the experiments with the text data that has been preprocessed in the form of vectors and the specific parameters before starting, the results show that GRU achieves the highest performance value in accuracy, recall, precision, and F1 score. Although GRU becomes the model with the smallest file size, it is the slowest model to generate predictions from text news data. It also has a higher potential to be an overfitted model due to parameters than a simple RNN.**

Index Terms—**Deep Learning Model, Political Hoax News Detection, Text Classification**

I. Introduction

COMMUNICATION has shifted after the mid-1990s. Facebook, Twitter, Instagram, and WhatsApp help people to share real-time information across networks. Online social networks are essential for communication and information exchange because they are easy to use, rapid, and cheap. Hence, most social media users get news online. Due to the rise of online social networks, the Internet is excellent for distributing fake news, such as false content, reviews, rumours, ads, political statements, satires, and more [1].

Fake news is widely acknowledged as one of the most significant challenges facing democracies, journalists, and economies in recent years [2]. The widespread use of fake news to confuse and persuade Internet users with skewed information has made it a severe worry for industry and academics. In addition, a vast amount of false and misleading material is manufactured and disseminated on the Internet, posing a threat to online social groups and devastatingly affecting Internet activities such as online shopping and social networking [3]. The spreading nature of false news impacts millions of individuals and their environments, making it difficult to detect and identify fake news on social media platforms [1]. For example, some political events, particularly the debatably close Brexit vote in the U.K. and Donald Trump's narrow victory in the U.S. presidential election of 2016 have sparked a surge of interest in the concept of "fake news", which is hugely thought to have significantly impacted the results of both political campaigns [4].

As fake news, hoaxes are frequently portrayed as the "dirty" side of politics to denigrate political opponents rather than provide valuable facts, depending more on feelings than logic. The broad use of false news to mislead and convince Internet users is also prevalent in Indonesia. In the 2019 Indonesian presidential and parliamentary elections, hoaxes spread on social media to create distinctions between the two presidential candidates (the Jokowi and Prabowo camps) [5].

However, political democracy requires the free flow of information and mainstream mass media journalism. Although hoaxes spread on social media are a severe

Received: Dec. 02, 2023; received in revised form: June 10, 2024; accepted: June 10, 2024; available online: Aug. 21, 2024.
*Corresponding Author

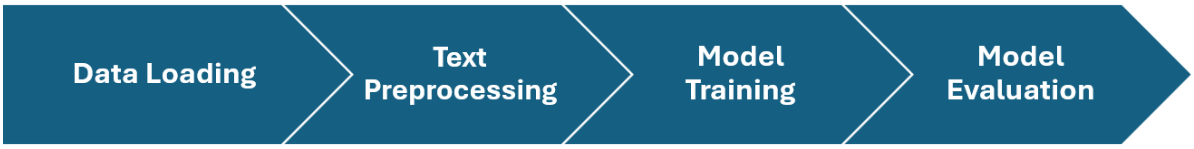


Fig. 1. The stages in the research.

threat to democracy, because the weight of the information spread cannot be measured, the speed at which it is delivered–without factual confirmation–turns it into a hoax that harms majority [6]. Furthermore, the impact of the spread of hoaxes has negative repercussions and hurt several parties. Hoaxes can also create losses from various perspectives, including time and money, public fear, and deterioration of social relationships, among others [7].

There are some of the literature studies that discuss Indonesian hoax news detection and proposed techniques that build into the system. First, previous research has compared several supervised text classification tasks, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Support Vector Machine (SVM), Naive Bayes (NB), and 1 Dimensional-Convolutional Neural Network (1D-CNN) and used Term Frequency-Inverse Document Frequency (TF-IDF) to eliminate the most common terms and extract only the most relevant terms from the corpus. The results indicate that 1D-CNN achieves the highest performance value of 97.9% even though 1D-CNN is well known to perform better in image and visual recognition due to its ability to capture semantic information of the text and its flexibility to classify larger datasets. The 1D-CNN model contains the tuned hyperparameters, including 128 filter size with five kernel size, default stride, Rectified Linear Unit (ReLU) activation function, and GlobalMaxPooling1D to sample down feature maps [8]. Second, another previous research proposes a hoax detection system using readers’ feedback and a text-matching approach using NB algorithm. It consists of four stages: pre-processing, similarity calculation, classification, and class determination. The results show that Naïve Bayes combined with probability-based feature selection of 0.2 achieves the best accuracy value of 0.87. Meanwhile, the other performance values, such as precision, recall, and f-measures, are 0.91, 1, and 0.95, respectively [9]. Third, previous research has compared supervised text classification tasks such as Multilayer Perceptron (MLP), NB, Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT) through the tokenizing process, case folding, normalization, filtering, stop words removing, stemming, and TF-IDF weighting using unigram and bigram features. The results indicates that RF algorithm has the highest performance value for accuracy and F1 score calculation of 76.47% and 74.24%, respectively [7].

Based on a review of related works, previous researchers have utilized supervised learning tasks to detect hoax or fact news on their datasets and conducted a performance comparison of model prediction to the test data split. Meanwhile, the research proposes applying supervised learning tasks in the form of several deep learning models to detect fake news in Indonesian political information equipped with an intimate text preprocessing phase. Thus, the research aims to eliminate fake news from a diffuse pool of information to reduce the quantity of misinformation because political news can be entirely made up and manipulated to gain attention as well as designed to mislead readers. The research also conducts supervised text classification tasks using various layers from different neural network types to clean the most common terms and release the most relevant terms from the corpus. To summarize, the major contribution of the research is the comparative analysis of basic layers in the deep learning model from different neural networks, with the aim of sorting news facts and hoaxes based on binary classification tasks. Furthermore, the research provides insights into the deep learning model usage in supervised learning tasks for text that completely with performance analysis during the training phase.

II. Research Method

The research has four main stages: retrieving data (data loading), text preprocessing, training model, and evaluating the model. Those stages have different actions to execute and various throughputs. The stages can be seen in Fig. 1

A. Data Loading

Data loading refers to taking data from a source file and importing it into a data frame type as a programming variable. The news data are obtained from the Kaggle site, entitled “Indonesian Fact and Hoax Political News”. It contains 41.726 rows of data from https://www.kaggle.com/datasets/linkgish/indonesian-fact-and-hoax-political-news.

TABLE I
DATASET DISTRIBUTION.

Label	Source	%
Fact	CNN, Kompas, Tempo	21.342
Hoax	Turnbackhoax	10.384

The dataset provides several columns, including title, timestamp, full text, tags, author, and URL. However, in the research, the researchers focus on processing data in the ‘full text’ column with the aim of hoax detection based on the article. Based on the data, news is gained from the famous news portal website. Meanwhile, the hoax news obtained from Turnbackhoax refers to a community website that fights the circulation of hoaxes in Indonesia. The website has been recognized by the Ministry of Communication and Information of the Republic of Indonesia. The dataset distribution is in Table I.

Next, a word cloud is a visual representation of a group of words that appear in a collection of text to facilitate understanding of the frequency and distribution of words in a text and identify the main topics contained in the text [10]. Words that appear more frequently in the text are displayed in a larger size and are more prominent in the word cloud. Meanwhile, words that occur less frequently are displayed in a smaller size.

Datasets containing news containing facts tend to describe many words from figures who play a role in politics. It can be observed in the representation of words such as “*kata*” and “Jokowi”. Meanwhile, datasets containing news containing false information tend to display comments from social media users. It is shown in the representation of words such as “*akun*” and “*sebut*”. With clearly visible differences in the distribution of the number of words spread across the dataset, it is easy for the model to give high weight to significant words in determining fact or hoax news. Figure 2 shows the word cloud.

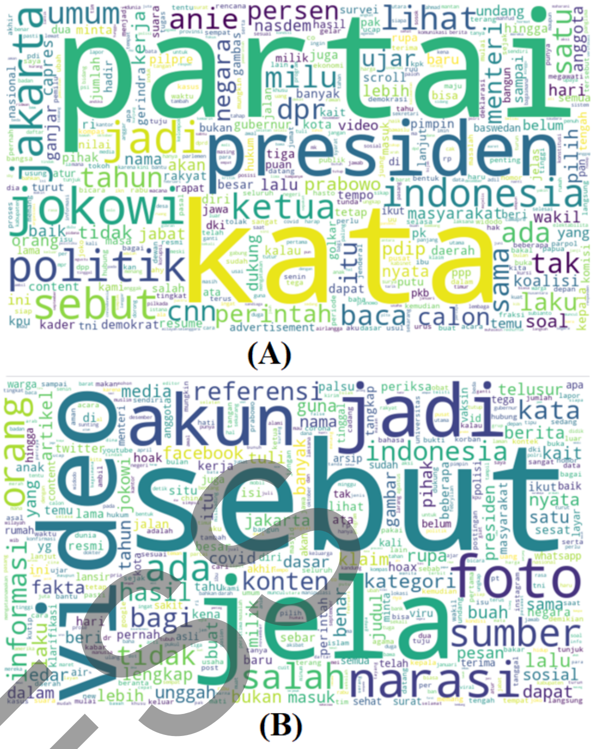


Fig. 2. Data of word cloud: (a) Word frequency of facts news data, (b) Word frequency of hoax news data.

TABLE II
EXAMPLE OF TEXT DATA TOKENIZATION PROCESS RESULTS.

Textual Raw Data	Tokenization
Jakarta, CNN Indonesia – Mantan Gubernur DKI Jakarta Anies Baswedan menghadiri acara Tasyakuran	‘jakarta’, ‘cnn’, ‘indonesia’, ‘*mantan*’, ‘gubernur’, ‘dki’, ‘jakarta’, ‘anies’, ‘baswedan’, ‘*menghadiri*’, ‘*acara*’,‘*tasyakuran*’

B. Text Preprocessing

Text preprocessing is an essential stage in the process of mining textual data that encompasses the purification and conversion of raw data into a structure that can be conveniently scrutinized by machine learning algorithms [11]. This stage aims to enhance the calibre of the textual information, optimize the effectiveness of subsequent analysis, and tackle obstacles associated with noise, incongruities, and fluctuations in the text [12]. The research uses five steps according to the system’s needs. The text preprocessing steps are broken down into the following details. First, tokenization frequently becomes the first step in various Natural Language Processing (NLP) tasks, especially in text classification. This step removes punctuation, symbols, and numbers as nonalphabetic characters from raw textual data. Tokenization aims to create meaningful units that can be further processed called tokens, which can be words, phrases, or other text elements [13]. Another preprocessing step is lowercasing all letters in all words, with the objective of data normalization, which reduces the potential for inconsistencies caused by the words or phrases that are capitalized differently in the original text being treated as the same word [13]. The example of text preprocessing can be seen in Table II.

Second, stop words are so prevalent in a corpus that their presence can become uninformative. Stop words frequently get eliminated during open vocabulary text mining preprocessing, and removing them decreases

TABLE III
EXAMPLE OF STOP WORDS REMOVAL PROCESS RESULTS.

Textual Raw Data	Stop Words Removal
'*pembina*', 'gerindra', '*yang*', '*juga*', '*menteri*', '*pariwisata*', '*dan*', '*ekonomi*', '*kreatif*', 'sandiaga', 'uno', '*hingga*', '*menteri*', 'bumn'	'*pembina*', 'gerindra', '*menteri*', '*pariwisata*', '*ekonomi*', '*kreatif*', 'sandiaga', 'uno', '*hingga*', '*menteri*', 'bumn'

TABLE IV
EXAMPLE OF LEMMATIZATION PROCESS RESULTS.

Textual Raw Data	Lemmatization
'*ibu*', '*pengajian*', '*mewujudkan*', '*keberhasilan*', '*pendidikan*', '*keluarga*', '*mengatakan*'	'*ibu*', '*aji*', '*wujud*', '*hasil*', '*didik*', '*keluarga*', '*kata*'

TABLE V
EXAMPLE OF TERM-FREQUENCY PROCESS RESULTS.

Word	Term-Frequency Process
'*ibu*'	4
'*kata*'	5
'jakarta'	2

computation time [14]. The elimination of stop words is a commonly accepted principle in text preprocessing. However, it cannot be considered a definitive mandate. The example is in Table III.

Third, lemmatization is a text preprocessing technique that involves removing word suffixes to transform them into their root form, called a lemma. Reducing words to their lemmas helps to achieve a better normalization and maintain the semantic integrity of the text [14]. For instance, the lemmatization process converts words like "running", "runs", and "ran" to the lemma "run". The example can be seen in Table IV.

Fourth, term frequency is a text preprocessing technique that counts the number of times a term appearing in a document. It is a fundamental statistic that provides insights into the relevance of a term within a document, including keyword extraction, document ranking, and information retrieval [15]. Meanwhile, TF-IDF calculates the importance score of a term in a collection of documents. The example can be seen in Table V.

Fifth, token filtering refers to the text preprocessing of eliminating tokens (i.e., words or phrases) from a text corpus based on their frequency of occurrence that does not contribute significantly to the analysis. Setting a threshold simplifies the document's representation and improves the efficiency and effectiveness of text-mining algorithms. Equation (1) shows ounting the number of words (w_i) in the document (d) and transforms it into frequency matrix (F_d). After that, the researchers select words as keywords by the threshold of their frequency (th_d) that describes the half of maximum term frequency score, as shown in Eq. (2). Then, the researchers apply a filter threshold (th_d) to the term frequency matrix (F_d) to reduce the number of terms. It is formulated in Eq. (3).

$$F_d = \lfloor f_{w_i d} \cdots f_{w_n d} \rfloor , W \in d, \tag{1}$$

$$th_d = \frac{1}{2} \max\{F_d\}, \tag{2}$$

$$F_d = \{ f_{w_i d} | f_{w_n d} \geq th_d \} , w \in d. \tag{3}$$

Sixth, data normalization transforms data into a standard scale or range and eliminates inconsistencies and variations to improve the accuracy and effectiveness of further steps. It aims to bring different features or variables onto a comparable scale, enabling fair and meaningful comparisons between them [16]. The researchers choose Z-Score (as shown in Equation (4)) to normalize term frequency score after token filtering by threshold because it standardizes features by transforming them to have a mean of 0 (shown in Eq. (5)) and a standard deviation of 1 (shown in Eq. (6)). It ensures that the features are on a comparable scale and prevents the dominance of certain variables based on their original scale or the disproportionate influence of the results [17].

$$z = \frac{x - \mu}{\sigma}, \tag{4}$$

$$\mu = \frac{1}{n} \sum_{i=0}^{n} x_i, \tag{5}$$

$$\sigma = \sqrt{\frac{i}{n} \sum_{i=0}^{n} (x_i - \mu)^2}. \tag{6}$$

C. Model Training

During this phase, the model is optimized to minimize a loss function. It measures the difference between its predicted output and the actual output by learning the underlying patterns to make accurate predictions on data with unrecognized patterns [18]. Especially for deep learning models, they must be trained to optimize their performance on a labelled dataset because the model adjusts its internal parameters to learn complex representations and hierarchical features from the data [19]. The researchers train three different types of simple deep learning models to discover the most efficient and effective layer in building a high-accuracy Indonesian political fake news detection system. The layers are LSTM, 1DCNN, and GRU to handle the text classification in this case. The researchers split data into stratified random train and test subsets to

TABLE VI
DATA SPLIT NUMBERS.

Subset	Numbers	Percentage
Train	28.182	90%
Test	3.132	10%

TABLE VII
HYPERPARAMETERS USED IN THE RESEARCH.

Hyperparameters	Value
Optimizer	Adam
Learning rate	0.001
Loss function	binary_crossentropy

understand how well the deep learning model performs and compares its performance against other models or benchmarks. Splitting the data also ensures that the model is evaluated on independent data, which can obtain a more accurate estimation of its generalization ability in real-world scenarios. The data split numbers are in Table VI.

Deep learning models typically require much data to achieve optimal performance [20]. K-fold cross-validation uses available data better by training the model on different training and validation subset combinations. This method helps to mitigate the risk of overfitting because the model is exposed to a greater variety of training instances. K-fold cross-validation ensures that each fold represents a diverse portion of the data. Each fold likely contains a proportional representation of different classes or data characteristics, providing a more balanced assessment of the model's performance [21].

D. Model Performance Analysis

The model performance analysis involves analyzing the model's predictive or classification accuracy using the Confusion matrix and gaining insights into its behavior on testing subset data. The Confusion matrix offers a comparative measure for the classification outcome of the model's predictive performance [22]. Its usage aims to understand better the classification errors of the model's performance [23]. The researchers use a Confusion matrix as the primary objective of model performance analysis because it identifies specific error types and potential class imbalances with understanding the distribution of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). From the Confusion matrix, various evaluation metrics can be derived to assess the model's performance based on the result of actual and prediction labels.

The accuracy metric provides an intuitive measurement by calculating the proportion of correctly classifying instances across all classes that represents the percentage F1-error rate. It is a standard evaluation metric used in data mining study cases because it reduces classifier performance to a single number and is easy to calculate [24]. The formula can be seen in Eq. (7).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{7}$$

The precision metric is measured by quantifying the total number of correct optimistic predictions compared to the total number of positively predicted data. However, achieving a high precision value does not necessarily ensure commendable overall model performance, particularly when the model prioritizes precision over recall [22]. The formula can be seen in Eq. (8).

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{8}$$

The recall metric measures the accuracy of the model in identifying positive instances and irrespective of the number of false positives. It quantifies the proportion of correctly identified positive instances [25]. A higher recall value denotes a reduced incidence of false negatives, signifying the efficacy of the model in detecting positive instances and minimizing the likelihood of overlooking them. The formula can be seen in Eq. (9).

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{9}$$

F1 score calculates the average prediction balance, especially for unbalanced datasets. So, it contains a formula that combines precision and recall [26]. F1 score has a value range between 0 and 1. The values close to 1 indicate almost perfect model performance. Meanwhile, values close to 0 indicate poor model performance. The formula can be seen in Eq. (10).

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{10}$$

III. RESULTS AND DISCUSSION

In deep learning models, hyperparameters are adjustable parameters that must be set before the training process begins. Examples of hyperparameters in a deep learning model include the learning rate, number of hidden layers, number of neurons per layer, activation function, and optimizer. In Table VII, the researchers choose 'adam' for the optimizer parameter before training the model due to its effectiveness in training neural networks that combine the benefits of adaptive learning

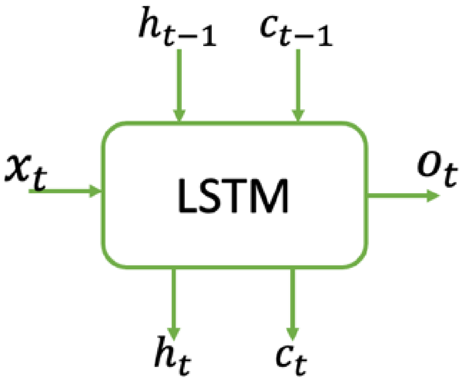


Fig. 3. Structure of Long-Term Short Memory (LSTM) cell.

rates and momentum-based updates, leading to faster convergence and improved performance. The Adam optimizer is also a common choice for binary classification in deep learning. Meanwhile, binary cross-entropy measures the dissimilarity between predicted probabilities and accurate binary labels, making it well-suited for models that classify inputs into one of two classes. Binary cross-entropy is a loss function which is suitable for specific tasks in data binary classification. Then, a learning rate of 0.001 as a starting point to train a deep learning model represents a moderate step size for weight updates. The value is insignificant to cause unstable training and slow down convergence.

The following are the experiments the researchers conduct in analyzing the three different types of layers in the deep learning model. In this case study, the analysis of model evaluation can be used to decide on the layer of deep learning models that can build an Indonesian political fake news system. It compares the models’ performance against a single scalar value and making an informed selection based on the specific requirements.

First, LTSM is a variant of Recurrent Neural Networks (RNN) architecture commonly used for text classification [27]. LSTM model can remember relevant information over longer sequences while also considering the more recent context by utilizing recurrent connections and specialized memory cells that selectively retain information over time [28].

In an LSTM cell, there are extra gates, namely the input, forget, and output gates that decide which signals are forwarded to another node [29]. Figure 3 shows the input and outputs of an LSTM cell process for a single timestep. This layer means that these equations have to be recomputed for the next time step. Thus, if the researchers have a sequence of 10 timesteps, the previously mentioned equations will be computed ten times for each timestep.

Based on Fig. 3, the input (x_t) is the incoming data at time step (t), such as words in a sentence. The hidden state (h_t) is a vector carrying information from the previous time step, used to influence the decision in the time step. Cell state (c_t) is a vector that carries long-term information through the sequence, which is updated by combining old information ($c_t - 1$) and new input through the gate. The output (o_t) is generated from the hidden state after the cell state has been processed and represents the information retained and processed by the LSTM at that time step. Overall, the input, hidden, and cell states work together to produce an output based on current and previous information.

The LSTM has an input x_t which can be the output of a Convolution Neural Network (CNN) or the input sequence directly. The h_{t-1} and c_{t-1} are the inputs from the previous timestep of LSTM. Moreover, o_t is the output of the LSTM for this timestep. The LSTM also generates the c_t and h_t for the consumption of the next time step in LSTM. The LSTM equations (Eqs. (11)–(16)) also generate f_t, i_t, and c'_t. These are for internal consumption of the LSTM and are used for generating c_t and h_t. The weight matrices ($W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c$) and biases ($b_f, b_i, b_o, b_c$) are not time-dependent. It means that these weight matrices do not change from one time step to another.

$$f_t = \sigma_g = (W_f \times x_t + U_f \times h_{t-1} + b_f), \quad (11)$$

$$i_t = \sigma_g = (W_i \times x_t + U_i \times h_{t-1} + b_i), \quad (12)$$

$$o_t = \sigma_g = (W_f \times x_t + U_o \times h_{t-1} + b_o), \quad (13)$$

$$c'_t = \sigma_c = (W_c \times x_t + U_f \times h_{t-1} + b_c), \quad (14)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t, \quad (15)$$

$$h_t = o_t \cdot \sigma_c(c_t). \quad (16)$$

In this experiment, the researchers create a simple LSTM model for Indonesian political fake news detection. It consists of four layers: InputLayer, Embedding, LSTM, and Dense. Setting 64 as the output dimension parameter for the LSTM layer aims to enable faster training and inference without excessively sacrificing performance. Figure 4 shows LSTM model plot.

Second, 1D CNN is a variant of neural network architecture used in deep learning to process sequential data, such as time series, text data, or grayscale images [30]. A 1D CNN model comprises one or more one-dimensional convolutional filters involving non-linear activation functions, pooling layers, and fully connected layers [31].

Based on Fig. 5, text data is represented as a sequence of features (x_0, x_1, x_2, and x_3), such as word vectors. The kernels (k_0, k_1, and k_2) are filters that perform convolution by shifting along the data

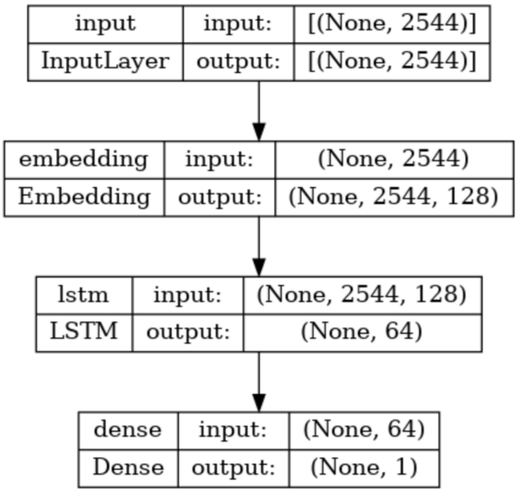

input | input: | [(None, 2544)]
InputLayer | output: | [(None, 2544)]
embedding | input: | (None, 2544)
Embedding | output: | (None, 2544, 128)
lstm | input: | (None, 2544, 128)
LSTM | output: | (None, 64)
dense | input: | (None, 64)
Dense | output: | (None, 1)

Fig. 4. Long-Term Short Memory (LSTM) model plot.

sequence, multiplying and summing the data elements to produce convolution maps (c_0 and c_1). The result of this convolution operation, namely the convolution map, is processed through an activation function such as ReLU, which removes negative values and retains only positive ones.

In the 1D CNN feature extractor, the convolution layer uses x as the input value [32]. Each node of c_i in the convolution layer is defined in Eq. (17), where k is a kernel (also called filter), and l is the number of kernels used. The output of c_i is an input to an activation function ReLU defined in Eq. (18). The activation results constitute a convolution map. These procedures of the convolution layer are shown in Fig. 5.

$$c_i = \sum_{j=0}^{n} k_j \times x_{i+j} + b, \tag{17}$$

$$ReLU(c_i) = \max(0, c_i), \tag{18}$$

In this experiment, the researchers create a simple 1D CNN model for Indonesian political fake news detection consisting of four layers: InputLayer, Embedding, Conv1D, GlobalMaxPooling1D, and Dense. The GlobalMaxPooling layer aggregates the most salient features by taking the maximum value across each feature map to summarize the most critical features globally and ensure that the model’s predictions are invariant to small translations in the input data. Figure 6 shows 1D CNN model plot.

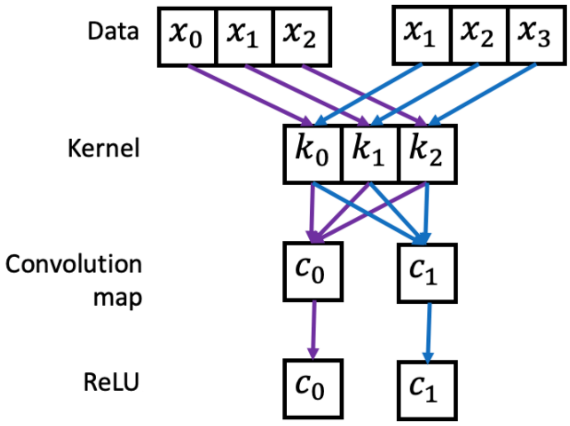


Fig. 5. Structure of One-Dimensional Convolutional Neural Network (1D CNN) cell.

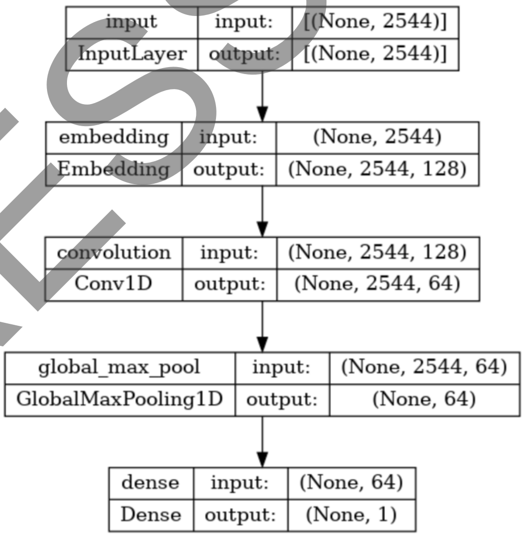

input | input: | [(None, 2544)]
InputLayer | output: | [(None, 2544)]
embedding | input: | (None, 2544)
Embedding | output: | (None, 2544, 128)
convolution | input: | (None, 2544, 128)
Conv1D | output: | (None, 2544, 64)
global_max_pool | input: | (None, 2544, 64)
GlobalMaxPooling1D | output: | (None, 64)
dense | input: | (None, 64)
Dense | output: | (None, 1)

Fig. 6. One-Dimensional Convolutional Neural Network (1D CNN) model plot.

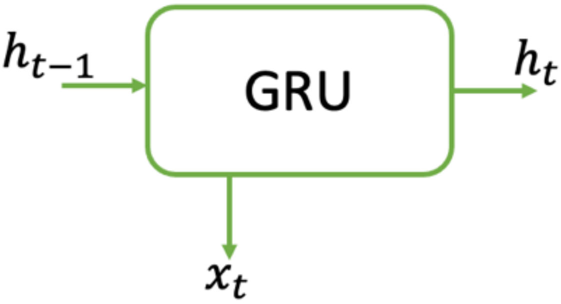


Fig. 7. Structure of Gated Recurrent Unit (GRU) cell.

Third, GRU is a variant of RNN with a similar architecture to the LSTM, but it has fewer parameters and is computationally less expensive. The GRU architecture has two gates. The reset gate determines how much of the previous hidden state should be forgotten. Then, the update gate determines how much of the new input should be added to the current hidden state [33]. Figure 7 shows structure of GRU cell.

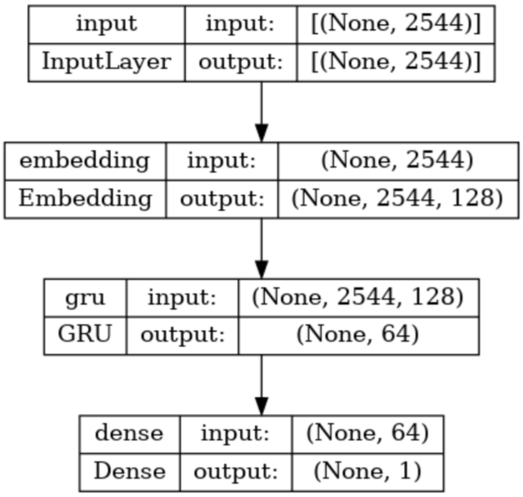


Fig. 8. Gated Recurrent Unit (GRU) model plot.

Based on Fig. 7, the hidden state ($h_t - 1$) carries information from the previous time step, while the input (x_t) is the incoming data at the current time step, such as words or features in the text. The output of GRU is the new hidden state (h_t), which carries the updated information from the current time step to the next time step. GRU regulates the flow of information by determining how much information from the previous hidden state should be forgotten and how much new information from the current input should be added to the hidden state.

The quantity of u_t is a gate vector. Recall the sigmoid function switches sharply between one and zero. So, when u_t is one, h is just a copy of the old h. The researchers ignore the input x since it is based on the value c_t. The gate r_t determines how much of the old state goes into defining the value of c_t.

$$r_t = \sigma(W_r X_t + U_r h_{t-1} + b_r), \qquad (19)$$

$$u_t = \sigma(W_u X_t + U_u h_{t-1} + b_u), \qquad (20)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \qquad (21)$$

$$c_t = \tanh(W x_t + U(r_t \cdot h_{t-1}) + b), \qquad (22)$$

$$h_t = u_t \cdot h_{t-1} + (1 - u_t) \cdot c_t. \qquad (23)$$

In this experiment, the researchers create a simple GRU model for Indonesian political fake news detection consisting of three layers, including four layers: InputLayer, Embedding, GRU, and Dense (Fig. 8). GRU consumes less memory than LSTM because it has no separate memory cell. Thus, it can be beneficial when dealing with resource-constrained environments that require parallelization across multiple devices [34].

Based on the theoretical review of GRU, LSTM, and 1D CNN, these three neural network architectures have their characteristics and advantages in handling text classification tasks. The researchers summarize the analysis of three neural network criterias in Table VIII. The researchers can inspect the number of parameters required by each architecture to function effectively. The number of parameters in a model plays an essential role in determining the complexity and size of the resulting model.

TABLE VIII
Criteria Comparison.

Criteria	GRU	LSTM	ID CNN
Type	Recurrent	Recurrent	Convolution
Amount of Parameter	$2n^2 + 3n$	$4n^2 + 4n$	$(k \times f \times d) + b$

Note: One-Dimensional Convolutional Neural Network (1D CNN), Long-Term Short Memory (LSTM), and Gated Recurrent Unit (GRU).

GRU has fewer parameters than LSTM, with the formula $2n^2 + 3n$, where n is the number of units in one GRU layer. The $2n^2$ component comes from the two weight matrices connecting the input and hidden states, while $3n$ comes from the bias and the three main gates in GRU, including reset, update, and new gate.

On the other hand, LSTM, has a more complex architecture with a larger number of parameters, namely $4n^2 + 4n$, where n is the number of units in one layer of the LSTM. The $4n^2$ component comes from the four main weight matrices corresponding to the input, forget, cell state, and output gate. Meanwhile, $4n$ comes from the bias and the gates.

For a 1D CNN, the number of parameters depends on the number of filters f, filter length k, and input dimension d. The formula for the number of parameters is $(k \times f \times d) + b$, where b is the bias. The $k \times f \times d$ component represents the number of parameters in the filter that convolves the input, and b is the bias added to the convolution result. The number of parameters in 1D CNN is usually less than that of GRU and LSTM, especially if the filter length k and the number of filters f are moderately set.

A. Discussion

The researchers monitor the performance history of the model process on both the training and validation datasets to observe how the model's performance evolves during the training phase, whether it converges and exhibits any signs of overfitting or underfitting. The researchers visualize the model's progress in the form of loss and accuracy values over epochs iterations. Then, a callback is a function called repeatedly during a process that validates or corrects certain

TABLE IX
EARLY STOPPING PARAMETERS.

Parameter	Value
monitor	‘val los’
patience	2

TABLE X
LEARNING RATE ADJUSTMENT PARAMETERS.

Parameter	Value
monitor	‘val los’
factor	0.4
minimum	1×10^{-6}
patience	1

TABLE XI
MODEL CHECKPOINT PARAMETERS.

Parameter	Value
monitor	‘val los’
model	‘min’
save_best_only	true

TABLE XII
MODEL SPECIFICATIONS IN THE RESEARCH.

Parameter	Model		
	ID CNN	LSTM	GRU
Saved file (MB)	4.6	4.5	4.4
Model Prediction speed	12	31	32

Note: One-Dimensional Convolutional Neural Network (1D CNN), Long-Term Short Memory (LSTM), and Gated Recurrent Unit (GRU).

behaviours. In machine learning, the researchers can use callbacks to define what happens before, during, or at the end of a training epoch. Callbacks are special utilities or functions that are executed during training at given stages of the training procedure. They can prevent overfitting, visualize training progress, debug the code, and save checkpoints. The following are the definition of the callbacks used in these experiments with the set parameters.

First, early stopping is a callback used while training neural networks, which provides the advantage of using many training epochs and stops the training once the model’s performance stops improving on the validation dataset [35]. Table IX shows the parameters.

Second, learning rate adjustment is a callback that monitors a quantity. If no improvement is seen for a ‘patience’ number of epochs, the learning rate is reduced [36]. The researchers reduce the learning rate when a metric has stopped improving. Once the learning stagnates, models often benefit from reducing the learning rate by a factor of 2^{-10}. Table X shows the parameters.

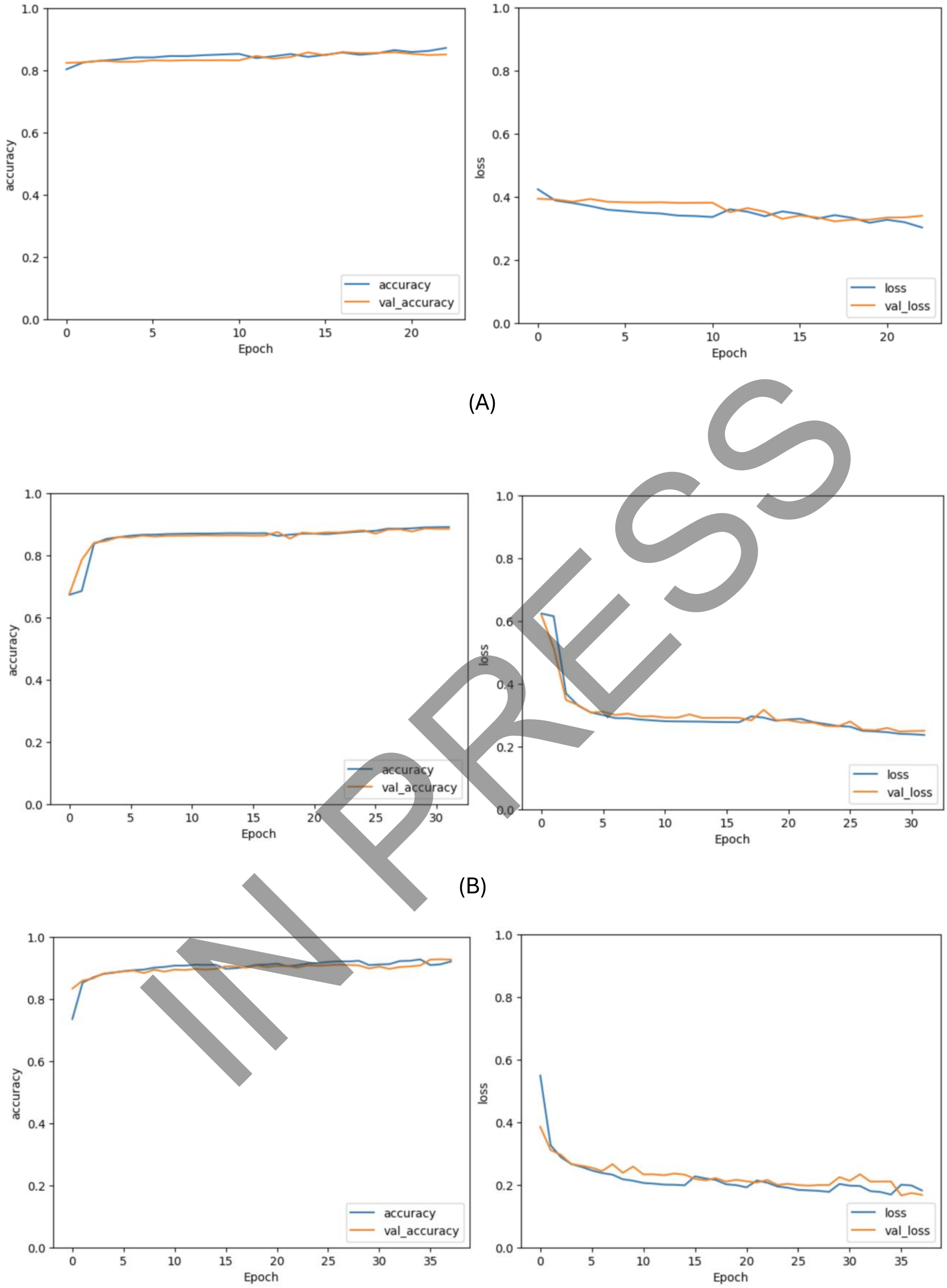
Third, checkpoints are snapshots of the working model during training, stored in non-volatile memory. In machine learning and deep learning experiments, they are essential used to save the current state of the model so one can pick up where one left off. Checkpoints capture the exact value of all parameters used by a model. They do not contain any description of the computation defined by the model and are typically only useful when source code that will use the saved parameter values is available [37]. Table XI shows the parameters.

Figure 9 shows the difference in the number of iterations during the learning phase due to the use of a callback called early stopping. Each model has unique behaviors in learning to classify the text data. Observation of the three accuracy and loss graphs shows that the learning process carried out by all models shows insignificant progress. However, there is a convergence between the training and validation datasets, although K-Fold is used. The classification model tends to perform insignificantly when validated using K-Fold due to imbalanced datasets and overfitting/underfitting of the model. Based on Figure 9, 1D CNN is a deep-learning model for text classification. It is the fastest in completing its learning because it cannot optimize the smaller loss value during the last few iterations.

Next, the researchers compare the specifications of each model to determine which produces the most implementable system, making it easy to apply in an application later. The researchers observe the model specification in two aspects. The aspects are the size of the saved file and the speed of the model in predicting the class label of the data.

Table XII shows that 1D CNN has the largest file size compared to other deep learning models for text classification because 1D CNN has different weights for each kernel. In addition, 1D CNN does not have a forget gate that LSTM and GRU own. However, 1D CNN has the fastest computational speed in predicting the label class of data because 1D CNN runs two equations shown in Eqs. (17)–(18). Meanwhile, LSTM and GRU have a more complex number of equations.

Next, the researchers compare performance metrics between models using a module called classification report from the scikit-learn library. The researchers look at accuracy, precision, recall, and F1 score to observe the performance of the tested deep-learning models. The classification report aims to assess the model’s performance for each class, identifying potential imbalances or biases in predictions. Thus, the researchers can understand the model’s performance against a single scalar value and make comparisons



(A)

(B)

(C)

Fig. 9. Training history plots: (a) One-Dimensional Convolutional Neural Network (1D CNN), (b) Long-Term Short Memory (LSTM), (c) Gated Recurrent Unit (GRU)

TABLE XIII
THE PERFORMANCE METRICS OF MODELS.

Metrics	ID CNN	LSTM	GRU
Accuracy	0.83	0.89	0.90
Precision	0.81	0.89	0.90
Recall	0.80	0.86	0.87
F1 Score	0.81	0.88	0.88

Note: One-Dimensional Convolutional Neural Network (1D CNN), Long-Term Short Memory (LSTM), and Gated Recurrent Unit (GRU).

across different models or variations.

Table XIII shows the values based on macro average calculation because the researchers need to treat all classes equally to evaluate the overall performance of the classifier against the most common class labels. Macro averaging gives equal weight to each category while micro averaging gives equal weight to each sample. For instance, the macro-averaged F1 score is computed using the arithmetic of all the per-class F1 scores. This method treats all classes equally regardless of their support values.

Based on Table XIII, the researchers can conclude that GRU is the model that has the best performance for fake Indonesian political news detection systems compared to LSTM and 1D CNN. However, GRU has the longest computation time in predicting the label class in the data. On the other hand, GRU executes as many as four equations, which are more straightforward than LSTM, as shown in Eqs. (19)–(23). Although GRU performs well compared to other types of deep learning layers in performing the binary classification task in this case study, it is actually more prone to overfitting because it has more parameters than a simple RNN. It is why GRU learns very well with the given training data but fails to recognize patterns from new data.

IV. CONCLUSION

The researchers compare different types of deep learning models for the binary text classification task using preprocessed text data in vectors. The 1D CNN, LSTM, and GRU are compared by observing the learning history graph, inspecting model specifications, and evaluating the performance metrics. GRU is the model that has attracted the most attention. In the research, GRU requires the longest learning time but produces the lightest model file size and becomes the fastest model in predicting the label class of the data. Furthermore, GRU achieves the highest performance values in terms of accuracy, recall, precision, and F1 score.

The case study in the research is foundation to build an advance model to execute a binary classification model for detect hoax in Indonesian political news. Hence, if there is complex case study that requires multiclass labels, the research provides some basic knowledge to decide the layers used in text classification task. Through theoretical studies and observations of model performance during experiments, the researchers find that GRU is the most recommended architecture in the context of text classification, especially in the detection system of Indonesian hoax news.

While the research provides important insights into political hoax detection in Indonesia using deep learning models, some limitations need to be noted. The sample size is limited to data from a specific period, which may not fully reflect future hoax trends. In addition, the model used focuses on texts in Indonesian, so the results may need to be more generalizable to other language contexts.

The comparison conducted can inspire future research on selecting the best foundation layer in building a model for text classification using deep learning. A good foundation of a model will lead to performance that can be measured properly.

ACKNOWLEDGEMENT

The researchers would like to express their deepest appreciation to P3M PENS for their financial support.

AUTHOR CONTRIBUTION

Writing—original draft, O. C. R. R.; Methodology, O. C. R. R; Formal analysis, O. C. R. R., and Z. M. E. D.; Analysis result review, Z. M. E. D. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] S. R. Sahoo and B. B. Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” *Applied Soft Computing*, vol. 100, 2021.

[2] X. Zhou, R. Zafarani, K. Shu, and H. Liu, “Fake news: Fundamental theories, detection strategies and challenges,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. Melbourne VIC, Australia: Association for Computing Machinery, Feb. 11–15, 2019, pp. 836–837.

[3] X. Zhang and A. A. Ghorbani, “An overview of online fake news: Characterization, detection, and discussion,” *Information Processing & Management*, vol. 57, no. 2, 2020.

[4] A. Gelfert, “Fake news: A definition,” *Informal Logic*, vol. 38, no. 1, pp. 84–117, 2018.

[5] T. Duile and S. Tamma, "Political language and fake news: Some considerations from the 2019 election in Indonesia," *Indonesia and the Malay World*, vol. 49, no. 143, pp. 82–105, 2021.

[6] E. H. Susanto, "Social media, hoax, and threats against diversity in Indonesia," *International Journal of Innovation, Creativity and Change*, vol. 8, no. 12, pp. 328–344, 2019.

[7] T. T. Putri, S. HendryxWarra, I. Y. Sitepu, M. Sihombing, and Silvi, "Analysis and detection of hoax contents in Idonesian news based on machine learning," *Journal of Informatic Pelita Nusantara*, vol. 4, no. 1, 2019.

[8] B. P. Nayoga, R. Adipradana, R. Suryadi, and D. Suhartono, "Hoax analyzer for Indonesian news using deep learning models," *Procedia Computer Science*, vol. 179, pp. 704–712, 2021.

[9] B. Zaman, A. Justitia, K. N. Sani, and E. Purwanti, "An Indonesian hoax news detection system using reader feedback and Naïve Bayes algorithm," *Cybernetics and Information Technologies*, vol. 20, no. 1, pp. 82–94, 2020.

[10] K. Padmanandam, S. P. V. D. S. Bheri, L. Vegesna, and K. Sruthi, "A speech recognized dynamic word cloud visualization for text summarization," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. Coimbatore, India: IEEE, Jan. 20–22, 2021, pp. 609–613.

[11] M. J. Denny and A. Spirling, "Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it," *Political Analysis*, vol. 26, no. 2, pp. 168–189, 2018.

[12] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text mining in big data analytics," *Big Data and Cognitive Computing*, vol. 4, no. 1, pp. 1–34, 2020.

[13] M. T. F. Al Islami, A. R. Barakbah, and T. Harsono, "Social media engineering for issues feature extraction using categorization knowledge modelling and rule-based sentiment analysis," *JOIV: International Journal on Informatics Visualization*, vol. 5, no. 1, pp. 83–93, 2021.

[14] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text preprocessing for text mining in organizational research: Review and recommendations," *Organizational Research Methods*, vol. 25, no. 1, pp. 114–146, 2022.

[15] M. Alfian, A. R. Barakbah, and I. Winarno, "Indonesian online news extraction and clustering using evolving clustering," *JOIV: International Journal on Informatics Visualization*, vol. 55, no. 3, pp. 280–290, 2021.

[16] A. Adeyemo, H. Wimmer, and L. M. Powell, "Effects of normalization techniques on logistic regression in data science," *Journal of Information Systems Applied Research*, vol. 12, no. 2, pp. 37–44, 2019.

[17] S. K. R. Koduru, "A comprehensive analysis of normalization approaches for privacy protection in data mining," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 8, no. 5, pp. 144–157, 2022.

[18] N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg, "Unsupervised machine learning of topological phase transitions from experimental data," *Machine Learning: Science and Technology*, vol. 2, pp. 1–20, 2021.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[20] E. G. Adagbasa, S. A. Adelabu, and T. W. Okello, "Application of deep learning with stratified K-fold for vegetation species discrimation in a protected mountainous region using Sentinel-2 image," *Geocarto International*, vol. 37, no. 1, pp. 142–162, 2022.

[21] H. Ling, C. Qian, W. Kang, C. Liang, and H. Chen, "Combination of support vector machine and K-Fold cross validation to predict compressive strength of concrete in marine environment," *Construction and Building Materials*, vol. 206, pp. 355–363, 2019.

[22] W. M. Fatihia, A. Fariza, and T. Karlita, "CNN with batch normalization adjustment for offline hand-written signature genuine verification," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 1, pp. 200–207, 2023.

[23] P. Harrington, *Machine learning in action*. Simon and Schuster, 2012.

[24] F. Provost and T. Fawcett, *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc., 2013.

[25] F. Amin and M. Mahmoud, "Confusion matrix in binary classification problems: A step-by-step tutorial," *Journal of Engineering Research*, vol. 6, no. 5, 2022.

[26] D. Chicco and G. Jurman, "The advantages of the Matthews Correlation Coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, pp. 1–13, 2020.

[27] Y. Widhiyasana, T. Semiawan, I. G. A. Mudzakir, and M. R. Noor, "Penerapan Convolutional

Long Short-Term Memory untuk klasifikasi teks berita Bahasa Indonesia,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 4, pp. 354–361, 2021.

[28] A. Singh, S. K. Dargar, A. Gupta, A. Kumar, A. K. Srivastava, M. Srivastava, P. Kumar Tiwari, and M. A. Ullah, “[Retracted] Evolving long short-term memory network-based text classification,” *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1–11, 2022.

[29] S. Varsamopoulos, K. Bertels, and C. G. Almudever, “Decoding surface code with a distributed neural network–based decoder,” *Quantum Machine Intelligence*, vol. 2, pp. 1–12, 2020.

[30] H. Huan, Z. Guo, T. Cai, and Z. He, “A text classification method based on a convolutional and bidirectional long short-term memory model,” *Connection Science*, vol. 34, no. 1, pp. 2108–2124, 2022.

[31] T. Zhang and F. You, “Research on short text classification based on TextCNN,” in *Journal of Physics: Conference Series*, vol. 1757. IOP Publishing, 2021, pp. 1–7.

[32] S. Nam, H. Park, C. Seo, and D. Choi, “Forged signature distinction using convolutional neural network for feature extraction,” *Applied Sciences*, vol. 8, no. 2, pp. 1–14, 2018.

[33] A. Dutta, S. Kumar, and M. Basu, “A gated recurrent unit approach to Bitcoin price prediction,” *Journal of Risk and Financial Management*, vol. 13, no. 2, pp. 1–16, 2020.

[34] X. Liu, Z. Lin, and Z. Feng, “Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM,” *Energy*, vol. 227, 2021.

[35] S. Paguada, L. Batina, I. Buhan, and I. Armendariz, “Being patient and persistent: Optimizing an early stopping strategy for deep learning in profiled attacks,” *IEEE Transactions on Computers*, pp. 1–12, 2023.

[36] A. Thakur, M. Gupta, D. K. Sinha, K. K. Mishra, V. K. Venkatesan, and S. Guluwadi, “Transformative breast cancer diagnosis using CNNs with optimized ReduceLROnPlateau and early stopping enhancements,” *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, pp. 1–18, 2024.

[37] E. Rojas, D. Pérez, J. C. Calhoun, L. B. Gomez, T. Jones, and E. Meneses, “Understanding soft error sensitivity of deep learning models and frameworks through checkpoint alteration,” in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. Portland, USA: IEEE, Sept. 7–10, 2021, pp. 492–503.