

Helpdesk Ticket Classification for Technician Assignment Routes Using BiLSTM

Putu Alan Arismandika^{a1}, Kadek Yota Ernanda Aryanto^{a2}, I Made Gede Sunarya^{a3}

^aComputer Science Department, Graduate Program, Ganesha University of Education, Bali, Indonesia

e-mail: 1putu.alan.arismandika@gmail.com, 2yota.ernanda@undiksha.ac.id,
3sunarya@undiksha.ac.id

Abstrak

Sebagian besar bisnis proses perusahaan diakomodir oleh aplikasi. Seringkali aplikasi pada perusahaan mengalami masalah akibat faktor internal maupun eksternal. Masalah aplikasi tersebut dilaporkan melalui media helpdesk. Laporan masalah yang dilaporkan melalui helpdesk tidak langsung masuk kepada teknisi untuk penyelesaiannya melainkan masuk ke operator dan dieskalasi ke teknisi untuk diselesaikan. Proses tersebut berpengaruh pada efisiensi waktu penyelesaian masalah. Penelitian ini mengusulkan penggunaan klasifikasi teks dengan deep learning untuk menyelesaikan pekerjaan operator. Metode yang diusulkan dalam penelitian ini adalah metode BiLSTM. Total data yang digunakan dalam penelitian ini adalah sebanyak 160.000 data laporan masalah helpdesk dengan membagi data sebanyak 128.000 data resolved sebagai data training dan sebanyak 32.000 data on-progress sebagai data testing. Penelitian dilakukan menggunakan 13 label untuk proses rute penugasan teknisi. Pengujian hasil penelitian ini menggunakan confusion matrix yang mendapatkan nilai accuracy sebesar 91.18%, precision 95.05%, dan recall 93.28%.

Kata kunci: Klasifikasi Teks, Deep Learning, BiLSTM, Helpdesk

Abstract

Most of the company's business processes are supported by applications. However, these applications often experience problems due to various internal and external factors. When users encounter problems, they submit requests for help to the helpdesk system. Unfortunately, these requests do not go directly to the technicians but instead are first sent to an operator who must then escalate them to a technician. This process can delay problem-solving, reducing efficiency. To address this issue, this study suggests using text classification with deep learning to streamline the operator's work. Specifically, the proposed method uses BiLSTM. The study used a total of 160,000 helpdesk request data, dividing it into 128,000 resolved data for training and 32,000 on-progress data for testing. Thirteen labels were used to represent the route process. This study uses a confusion matrix to measure its performance. The results showed an accuracy of 91.18%, precision of 95.05%, and recall of 93.28%.

Keywords: Text Classification, Deep Learning, BiLSTM, Helpdesk

1. Introduction

The helpdesk system is a system that assists companies in developing and improving their service products. In public service companies, the helpdesk system is beneficial in enhancing services to its customers. Services that use the application often experience problems. Users request to solve the problem of the application by submitting it to the helpdesk system. The requests are then managed by several operators to be translated and grouped based on the problem category. The requests or the ticket problem category will be resolved by a technician who has the authority to solve the problem. The operator carries out the escalation process to the technician after translating and grouping the problems from the reports on the user services system. A large number of incoming problem reports makes the operator's workload high so that problems are often resolved longer than they should. In addition, operators also carry out escalation work manually by reading the problems that have been

reported one by one. Apart from requiring a long completion time, costs are also high due to having to recruit workers for more operator assignments.

The manual process for escalating requests from operators to technicians can adopt a text classification process. This process can be done with a deep learning approach. The incoming text is first processed and then classified based on technician class. The prediction function on text classification will recognize the class as the escalation destination to a technician who has the authority in solving the problem. The data used as training data in this study is BPJS Ketenagakerjaan helpdesk ticket report data for the period 2019 to May 2022. The test and prediction data used are data after June 2022 to December 2022. This data certainly has repeated text or frequently asked texts that can be used as a reference in the escalation process. The whole text classification process in the user services system requires a method used to classify text with high accuracy. In several studies of text classification with binary class and multiclass, the ability of RNN to have high accuracy in applying text classification compared to other classification methods such as CNN and other deep learning methods. For example, in the classification of sentiment analysis, the RNN method produces higher accuracy than CNN [1]. In several studies, many modifications of text classification have been carried out. In RNN there are several development methods including LSTM, Bi-LSTM, and GRU. In recent study, text classification is often applied to this development method to analyze comments about hotel visitors [2].

2. Study Method

The study method for helpdesk ticket classification using BiLSTM consists of 4 stages, namely study design, data collection, data processing, and model performance measurement.

2.1 Study Design

By the explanation in the introduction, the study design is made to assist in determining study planning, problem boundaries, and study can be completed according to schedule. The main objective of this study is to accommodate the operator's work as a route manager for helpdesk ticket reports to technicians with Bi-LSTM. In addition, this study also made predictions and tested the accuracy of BiLSTM. The conceptual framework for study flow outlines is the process of making the BiLSTM model based on training data as a reference for classification, input data testing up to the results of classification as a guideline for appointment or assignment of technicians. There are 13 labels used as a reference for the classification process, namely Data Pendukung, Executive Summary, Informasi Profile Kepesertaan, Kepesertaan Bukan Penerima Upah, Kepesertaan Jasa Konstruksi, Kepesertaan Penerima Upah, Keuangan, Network (Jaringan), Pelayanan, Pengawasan dan Pemeriksaan, Permintaan Antar Kantor (Wilayah), User dan Role, dan e-Channel. The conceptual framework is shown in Figure 1.

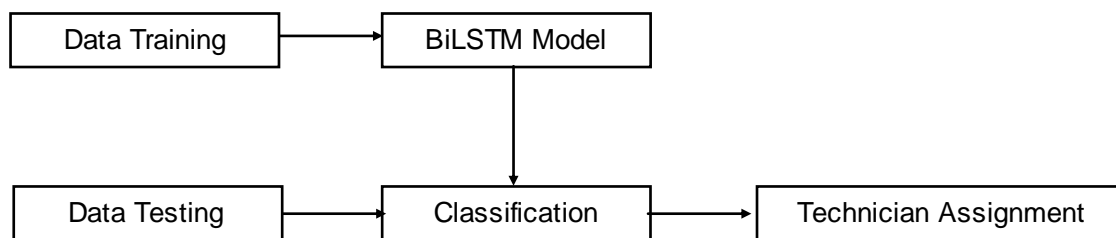


Figure 1 Study design concept flow

Data training is data that has been completed by the operator to be used as a model in BiLSTM. Data testing is data that becomes input test. Furthermore, the testing data is classified based on the BiLSTM model which is formed from training data. The resulting label of the classification will be the technician's assignment.

2.2 Data Collection

The total amount of data used in this study is 160,000 records. The data is BPJS Ketenagakerjaan helpdesk data for the period January 2019 to May 2022 of 128,000, which

have been completed by operators, and data for the period June 2022 to December 2022 of 32,000 data, which have not been completed by operators. Where the training data is data for the period January 2019 to May 2022 as many as 128,000, and testing data is data for the period June 2022 to December 2022 of 32,000. In percentage, the training data consisting of 128,000 records represents 80% of the total data or 80% of 160,000, while the testing data consisting of 32,000 records represents 20% of 160,000.

2.3 Data Preprocessing

Stages data pre-processing is carried out to process the data before it is used in the application of the Bi-LSTM method. Processing carried out in text processing techniques adjusts the state of the ticket report data helpdesk such as the presence of unnecessary characters and formatting template ticket helpdesk to then be classified in Bi-LSTM. The text processing technique used is case folding, remove number, remove tab and new line, remove character, stopword, your token, and word embedding.

2.4 Model Performance

To validate the correctness of the helpdesk ticket technician assignment route, in this study, a manual validation was used that was filled in by the operator as a validator to determine the technician assignment route process resulting from the BiLSTM classification process is true or false. In this case, the operator is the person who is working to escalate the assignment of helpdesk ticket technician manually. The validation form result is calculated with a confusion matrix. The validation results from the operator on the results from BiLSTM are calculated by FN, FP, TP, and TN to obtain the accuracy of the BiLSTM model.

3. Literature Study

3.1 Helpdesk System

Management of information systems and applications problems within a company or institution must be resolved quickly and precisely. These various problems can be accommodated and handled by implementing a reporting system. This reporting system is usually called a helpdesk system. Helpdesk is a single point for interaction between users and technicians, in this case, is the field of application development [3]. Interaction in the helpdesk is usually in the form of text input on a web-based form or a mobile smartphone application. Requests that have been submitted by the user are usually called helpdesk report tickets. In the helpdesk system, the ticket will be checked by a centralized team and forwarded to the relevant team to be resolved. [4]. Checks that are carried out manually by a centralized team are then identified to find out the problems that have been requested by the user. [5].

3.2 Deep Learning

In the last 10 years, students have been able to accomplish many different ideas in the application of machine learning. Deep learning is a sub-field of machine learning within machine learning. Machine learning is a part of AI and deep learning is a derivative of machine learning. Deep learning adopts the way the human brain works in processing data and making decisions. In various fields such as image processing, sound and video processing, language and text processing, and video games, deep learning has been applied, and until now the study carried out in this field has approached the human level [6]. The architectures and algorithms of deep learning are so large and varied that they have the potential to continue to be developed. In the last 20 years, there are 6 algorithms that have been developed to date. LSTM and CNN are the earliest developed methods that have high accuracy to solve various problems. CNN and RNN are part of Supervised Learning and SOM and Autoencoders are Unsupervised Learning category in deep learning. Each solution in the problem work, deep learning has its own best performance. For example, CNN is best in the field of image recognition and RNN for NLP [7].

3.3 Bidirectional LSTM

The development of the LSTM method is BiLSTM. In contrast to LSTM, BiLSTM performs data training twice. In LSTM, the data training model is done once and only gets one information, namely information on the previous process. [8]. BiLSTM uses the previous information and the information after it and processes it bidirectionally. BiLSTM has two layers,

namely the Layer and Backward Layer. The forward layer serves to get the previous information and the backward layer gets the information afterwards. In some studies, BiLSTM is widely used to obtain representations on data that requires repeated information. For example, in the representation of comment text. Bi-LSTM can accommodate the shortcomings of LSTM in obtaining information while representing information from data.

The comparison between the BiLSTM method and other methods such as LSTM, CNN, and RNN in text classification has many advantages. BiLSTM has the ability to access both preceding and subsequent information, and is effective in encoding long-distance word dependencies [9][10]. This means that information from distant words in a sentence can be captured based on their dependencies. In the case of long texts, some words at the beginning or in the middle of a sentence can affect the understanding of words at the end of the sentence. Bi-LSTM allows the model to take into account these long-distance relationships better than other methods such as CNN, RNN, and LSTM. In this study, the user's helpdesk request structure has a long sentence pattern, so BiLSTM is expected to effectively extract information from the request sentence.

3.4 Relevant Studies

Study studies relevant to text classification have been carried out by several studyers. Study by J. Zheng entitled A Novel Computer-Aided Emotion Recognition of Text Method Based on WordEmbedding and Bi-LSTM in 2019 states that the text in this study is converted from word features with Word Embedding. The purpose of this study is to classify emotions through word or text-based data. The accuracy result of the detection is above 64.09% [11]. Study by Yunsick Sung, Sejun Jang, Young-Sik Jeong, Jong Hyuk (James J.) Park with the title Malware classification algorithm using advanced Word2vec-based Bi-LSTM for ground control stations in 2020 contains the purpose of utilizing data preprocessing using fasttext for features in the Bi-LSTM method. In the study, the accuracy result was 96.76% or 0.76% greater than the previous study [12]. Study related to the comparison of deep learning methods for text classification has been conducted by Congcong Wang, Paul Nulty, and David Lillis in 2020. The study is entitled A Comparative Study on Word Embeddings in Deep Learning for Text Classification. In this study, studyers compared the performance of CNN and Bi-LSTM in text classification with the preprocessing stage, namely word embedding. The results of the study found that Bi-LSTM produced very good performance in text classification to get the context of the text [13]. Study in the application of text classification in helpdesk systems has also been carried out by M. A. Prihandono, R. Harwahu and R. F. Sari in 2020 with the study title Performance of Machine Learning Algorithms for IT Incident Management. The study produced the highest accuracy using the LSTM method, which was 98.86% compared to other deep learning methods [14]. Based on some relevant study, it can be understood that the RNN method with its development can be used in problems related to multiclass text classification-based predictions. In the study that has been mentioned, it produces high accuracy results for text classification. In addition, the helpdesk system also produces text classification with good results in the LSTM method. Due to some conclusions from the bibliography and previous study which states that the accuracy of BiLSTM is better than LSTM. Therefore, in this study, it is expected that BiLSTM can produce greater accuracy for technician assignment routes in helpdesk reports than previous study.

All relevant previous studies results show that the performance of BiLSTM is better than other methods such as LSTM, CNN, and RNN in text classification. Based on these findings, this study applies the BiLSTM method to improve the accuracy of the ticket escalation process from helpdesk to technicians.

4. Result and Discussion

4.1. Dataset Preprocessing

Data preprocessing serves to prepare raw data into data that is ready for processing and also as a basis for creating a BiLSTM model. The dataset will be converted into a sequence of numbers or numeric which aims to be a vector feature of each text and label in the classification process. The datasets that are processed are training data totaling 128,000 and testing data totaling 32,000. The dataset is processed using the same steps. The steps taken in this study are as follows. The first stage of data preprocessing is case folding. Case folding is the process of converting all uppercase letters into lowercase letters [15]. The purpose of this

process is so that computer machines can recognize each unique word and can categorize words that have the same meaning or in other words so that all data has the same form The function will convert all words and sentences from the helpdesk ticket report into lowercase letters.

The remove number process is a process to remove the numbers in the helpdesk report ticket. Numbers are removed to reduce unique characters in the tokenization process. The number characters are not needed in this study because they do not contain meaning for the escalation process or technician assignment. The number character removal process in Python is processed in the function below. The number character removal function will produce helpdesk report text data without numbers. The remove tab and new line process is a process that removes tabs and enters characters or removes excess spaces and new lines. This process functions so that the text can be easily converted into tokens and index words. The function removes excess spaces and new lines in the helpdesk report text that have been remove number before.

Remove Character is the process of removing characters other than letters and punctuation characters. The function removes all characters that are not needed in text processing, such as "@", "#" and "\$". The stopwords process is the process of removing conjunctions in Indonesian such as "di", "ke", "and", and so on. The process in Python programming is to use the help of the NLTK library which can call Indonesian stopwords. The process in the function above removes all conjunctions in the helpdesk ticket report text. The text processed in stopwords is the text that has been processed in the previous remove character. The snippet of text processing results is shown in Figure 2.

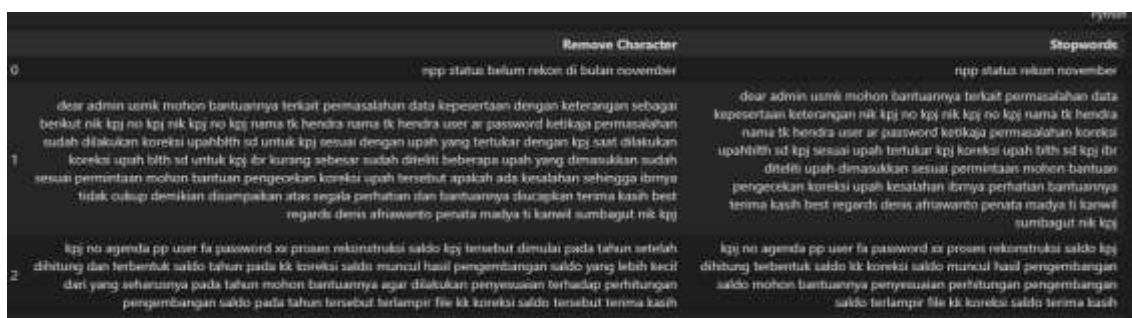


Figure 2 Stopwords

Tokenization is the process of converting sentences that have been processed by stopwords into word tokens, and then the word tokens are sorted based on the frequency of words in the sentence [16]. The word with the highest frequency will be sorted into the earliest index and continue until the word with the smallest frequency. After getting the frequency index of the word, the words in the sentence of the helpdesk ticket report will be converted to the frequency index number.

	Stopwords	Token
0	npp status rekon november	[32, 58, 320, 337]
1	dear admin usmk mohon bantuannya terkait permasalahan data kepesertaan keterangan nik kpj no kpj nik kpj no kpj nama tk hendra nama tk hendra user ar password ketikaja permasalahan koreksi upabhth sd kpj sesuai upah tertukar kpj koreksi upah bth sd kpj ibr diteliti upah dimasukkan sesuai permintaan mohon bantuan pengecekan koreksi upah kesalahan ibny perhatian bantuannya terima kasih best regards denis afriawanto penata madya ti kanwil sumbagut nik kpj	[4, 7, 11, 1, 2, 5, 3, 14, 74, 18, 21, 13, 38, 13, 21, 13, 38, 13, 16, 22, 948, 16, 22, 948, 52, 79, 59, 3, 41, 192, 13, 39, 90, 1785, 13, 41, 90, 91, 192, 13, 236, 90, 714, 39, 12, 1, 19, 108, 41, 90, 216, 9, 2, 8, 10, 15, 6, 3019, 3319, 37, 36, 47, 257, 1427, 21, 13]

Figure 3 Word Indexing

The padding process is designed to homogenize the shape of the long sequence of vectors generated from word indexing. In this study, the base value of the number sequence is the longest word, which is 550 words. Words that are less than 550 will be added with 0 at the front of the number vector sequence. The token sequence is added with 0 at the front of the sequence so that it has the same size of 550 number sequences. The number sequence will be processed last into text vector features in word embedding so that it can be processed in the BiLSTM model.

4.2. Classification

The classification process is a process where the feature vectors from the preprocessing and embedding processes are classified to be predicted with the testing data. After the feature vector is obtained by the word embedding process, the feature vector is then calculated with the Bi-LSTM method. That is by calculating the forget gate, input gate and output gate. The calculation of the forget gate, input gate, and output gate is done forward and backward on the feature vector generated from the calculation in word embedding. The calculation uses sigmoid and tanh. The feature vector will be sequentially be multiplied by the weight matrix and added with a bias. The initial weight and bias values in the Bi-LSTM model are initialized the same as word embedding using uniform glorot [17] [18]. In the BiLSTM classification process, the feature vectors from the training data are processed by forward and backward LSTM [19]. The feature vectors that have been calculated at the forget gate, input gate, and output gate will go through softmax activation to determine the value in performing classification. Softmax will take the largest probability as a prediction to determine the label.

The Bi-LSTM model created has an input dimension vector of 550, which is based on the highest number of word sequences and for the output, 256 is chosen as it was done in the embedding process. Model summary created is shown in Figure 4 and Figure 5.

```

Model: "sequential_10"
-----
Layer (type)                Output Shape              Param #
-----
embedding_10 (Embedding)    (None, 550, 256)         1194752
spatial_dropout1d_8 (SpatialDropout1D) (None, 550, 256)         0
bidirectional_7 (Bidirectional) (None, 256)               394240
dense_1 (Dense)              (None, 13)                3341
-----
Total params: 1,641,437
Trainable params: 1,641,437
Non-trainable params: 0
-----
None
    
```

Figure 4 Model Summary

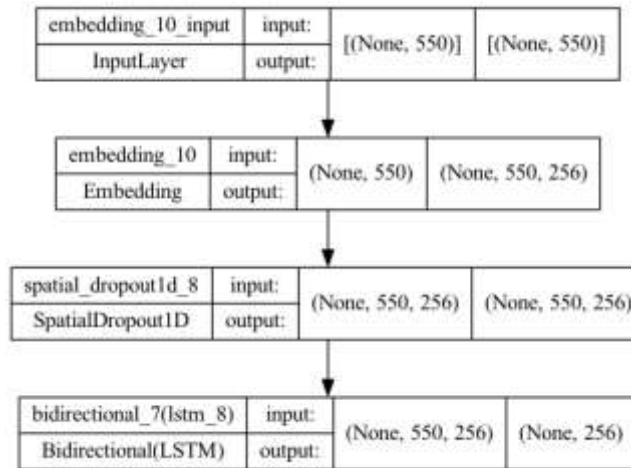


Figure 5 Model visualize

In the dense model used, there are 13 labels or classes in the classification. The label used is the field of technicians assigned to resolve helpdesk ticket reports. Feature vector data of 128,000 each has a label. Labels are converted into dummies or one-hot encode.

The model that has been designed previously will be measured by conducting a training process on the model to be able to classify and predict based on training data and training results. The training data used is a feature vector of 128,000 data with a vector dimension length of 550. The training process uses batch size scenarios of 64 and 128 and epochs of 10, 20, 50, and 100. A snapshot of the epoch display carried out in this study is shown in Figure 8, Figure 7, Figure 8, and Figure 9,

```
Epoch 1/10
1800/1800 [-----] - 6069s 3s/step - loss: 0.4838 - accuracy: 0.8299 - val_loss: 0.3445 - val_accuracy: 0.8788
Epoch 2/10
1800/1800 [-----] - 6208s 3s/step - loss: 0.3848 - accuracy: 0.8976 - val_loss: 0.3841 - val_accuracy: 0.8939
Epoch 3/10
1800/1800 [-----] - 6348s 4s/step - loss: 0.2682 - accuracy: 0.9184 - val_loss: 0.2913 - val_accuracy: 0.9827
Epoch 4/10
1800/1800 [-----] - 6414s 4s/step - loss: 0.2385 - accuracy: 0.9284 - val_loss: 0.2783 - val_accuracy: 0.9852
Epoch 5/10
1800/1800 [-----] - 6518s 4s/step - loss: 0.2078 - accuracy: 0.9278 - val_loss: 0.2769 - val_accuracy: 0.9878
Epoch 6/10
1800/1800 [-----] - 6407s 4s/step - loss: 0.1838 - accuracy: 0.9362 - val_loss: 0.2884 - val_accuracy: 0.9112
Epoch 7/10
1800/1800 [-----] - 6411s 4s/step - loss: 0.1653 - accuracy: 0.9424 - val_loss: 0.2932 - val_accuracy: 0.9116
Epoch 8/10
1800/1800 [-----] - 6570s 4s/step - loss: 0.1491 - accuracy: 0.9480 - val_loss: 0.3007 - val_accuracy: 0.9188
Epoch 9/10
1800/1800 [-----] - 6617s 4s/step - loss: 0.1332 - accuracy: 0.9538 - val_loss: 0.3153 - val_accuracy: 0.9188
Epoch 10/10
1800/1800 [-----] - 7881s 4s/step - loss: 0.1286 - accuracy: 0.9575 - val_loss: 0.3079 - val_accuracy: 0.9142
```

Figure 6 10 epochs

```
Epoch 6/20
1800/1800 [-----] - 6170s 3s/step - loss: 0.1823 - accuracy: 0.9368 - val_loss: 0.2888 - val_accuracy: 0.9182
Epoch 7/20
1800/1800 [-----] - 6181s 3s/step - loss: 0.1626 - accuracy: 0.9434 - val_loss: 0.3012 - val_accuracy: 0.9040
Epoch 8/20
1800/1800 [-----] - 6129s 3s/step - loss: 0.1467 - accuracy: 0.9492 - val_loss: 0.2992 - val_accuracy: 0.9097
Epoch 9/20
1800/1800 [-----] - 6128s 3s/step - loss: 0.1317 - accuracy: 0.9541 - val_loss: 0.3040 - val_accuracy: 0.9098
Epoch 10/20
1800/1800 [-----] - 6112s 3s/step - loss: 0.1179 - accuracy: 0.9596 - val_loss: 0.3340 - val_accuracy: 0.9126
Epoch 11/20
1800/1800 [-----] - 6117s 3s/step - loss: 0.1082 - accuracy: 0.9627 - val_loss: 0.3424 - val_accuracy: 0.9105
Epoch 12/20
1800/1800 [-----] - 7009s 4s/step - loss: 0.0962 - accuracy: 0.9662 - val_loss: 0.3429 - val_accuracy: 0.9182
Epoch 13/20
...
Epoch 19/20
1800/1800 [-----] - 6143s 3s/step - loss: 0.0688 - accuracy: 0.9796 - val_loss: 0.4368 - val_accuracy: 0.9130
Epoch 20/20
1800/1800 [-----] - 5877s 3s/step - loss: 0.0585 - accuracy: 0.9885 - val_loss: 0.4588 - val_accuracy: 0.9112
```

Figure 7 20 epochs

```

Epoch 6/50
1800/1800 [-----] - 7586s 4s/step - loss: 0.1813 - accuracy: 0.9372 - val_loss: 0.2842 - val_accuracy: 0.9107
Epoch 7/50
1800/1800 [-----] - 7341s 4s/step - loss: 0.1038 - accuracy: 0.9439 - val_loss: 0.2984 - val_accuracy: 0.9125
Epoch 8/50
1800/1800 [-----] - 7258s 4s/step - loss: 0.1462 - accuracy: 0.9490 - val_loss: 0.2910 - val_accuracy: 0.9110
Epoch 9/50
1800/1800 [-----] - 7225s 4s/step - loss: 0.1316 - accuracy: 0.9539 - val_loss: 0.3048 - val_accuracy: 0.9127
Epoch 10/50
1800/1800 [-----] - 7205s 4s/step - loss: 0.1184 - accuracy: 0.9588 - val_loss: 0.3248 - val_accuracy: 0.9119
Epoch 11/50
1800/1800 [-----] - 7211s 4s/step - loss: 0.1078 - accuracy: 0.9629 - val_loss: 0.3350 - val_accuracy: 0.9123
Epoch 12/50
1800/1800 [-----] - 7255s 4s/step - loss: 0.0976 - accuracy: 0.9660 - val_loss: 0.3410 - val_accuracy: 0.9110
Epoch 13/50
...
Epoch 49/50
1800/1800 [-----] - 6869s 4s/step - loss: 0.0326 - accuracy: 0.9886 - val_loss: 0.5878 - val_accuracy: 0.9189
Epoch 50/50
1800/1800 [-----] - 6912s 4s/step - loss: 0.0297 - accuracy: 0.9896 - val_loss: 0.5797 - val_accuracy: 0.9122
    
```

Figure 8 50 Epochs

```

Epoch 6/100
225/225 [-----] - 1591s 7s/step - loss: 0.2439 - accuracy: 0.9174 - val_loss: 0.3081 - val_accuracy: 0.8994
Epoch 7/100
225/225 [-----] - 1591s 7s/step - loss: 0.2389 - accuracy: 0.9288 - val_loss: 0.3017 - val_accuracy: 0.9085
Epoch 8/100
225/225 [-----] - 1677s 7s/step - loss: 0.2195 - accuracy: 0.9243 - val_loss: 0.2967 - val_accuracy: 0.9023
Epoch 9/100
225/225 [-----] - 1781s 8s/step - loss: 0.2085 - accuracy: 0.9281 - val_loss: 0.3001 - val_accuracy: 0.9035
Epoch 10/100
225/225 [-----] - 1771s 8s/step - loss: 0.2016 - accuracy: 0.9295 - val_loss: 0.2968 - val_accuracy: 0.9050
Epoch 11/100
225/225 [-----] - 1753s 8s/step - loss: 0.1910 - accuracy: 0.9326 - val_loss: 0.2972 - val_accuracy: 0.9038
Epoch 12/100
225/225 [-----] - 1750s 8s/step - loss: 0.1843 - accuracy: 0.9356 - val_loss: 0.3031 - val_accuracy: 0.9050
Epoch 13/100
...
Epoch 93/100
225/225 [-----] - 1558s 7s/step - loss: 0.0310 - accuracy: 0.9893 - val_loss: 0.6470 - val_accuracy: 0.9083
Epoch 94/100
225/225 [-----] - 1564s 7s/step - loss: 0.0325 - accuracy: 0.9889 - val_loss: 0.6433 - val_accuracy: 0.9097
Epoch 95/100
113/225 [----->.....] - ETA: 14:48 - loss: 0.0299 - accuracy: 0.9897
    
```

Figure 9 100 epochs

From the epoch scenario, all epochs produced an average accuracy of 91%. Where the results are shown in Table 1.

Table 1 Recapitulation of Training Model

Batch Size	Epoch	Accuracy
64	10	91.4%
64	20	91.1%
64	50	91.2%
64	100	91.1%
128	10	90.9%
128	20	91.2%
128	50	91.2%
128	100	91.1%

From the epochs carried out for epoch 10, linear results were obtained, which moved at 91% at epoch 8. While epochs 20, 50 and 100 occur starting at epoch 30.

Prediction is carried out on helpdesk data for the period June 2022 to December 2022 which has not been completed by the operator, which is 32,000 data. The data processing process passed by the testing data is the same as the training data, namely case folding, remove number, remove tab and new line, remove character, stopword, tokenization, and word embedding. The data processing is carried out to obtain feature vectors from the new helpdesk report data which will be used as testing data. Then the feature vector is predicted to get the appropriate label based on the testing data label. The prediction process is done with the

argmax function or determining the largest value of the label resulting from the Bi-LSTM calculation process. The media for testing the prediction of testing data used is a web-based application. The application is built using flask. An overview of the helpdesk ticket report technician assignment prediction media is shown in Figure 10.

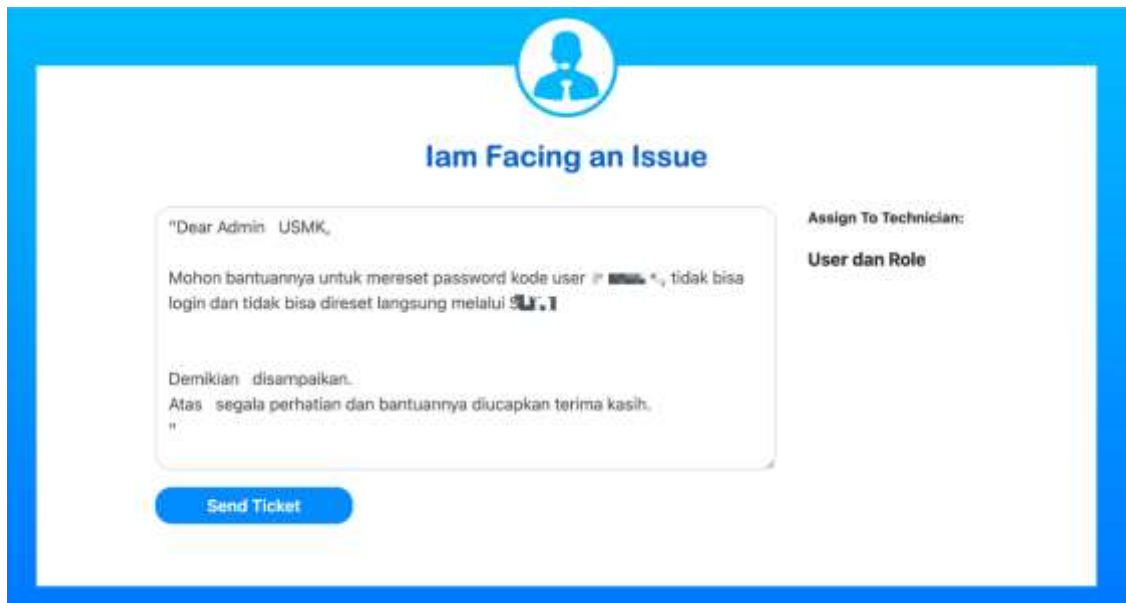


Figure 10 Application Interface

Based on the model that has been saved and applied in the flask web-based application, the results of the prediction will be saved into a .csv file and then manually validated by the operator. The suitability of the Bi-LSTM prediction will be compared with the actual prediction by the operator and given a "True" sign if the Bi-LSTM prediction and actual prediction produce the same prediction results and given a "False" sign if the prediction results are not the same. The overall results of the Bi-LSTM prediction and the actual prediction from the operator as well as the results of matching the prediction results are attached in the appendix of this study. Validation is done by the operator and approved by the operator coach. In this study, the results of the validation become the performance benchmark of the classification and prediction results of BiLSTM.

4.3 Performance Result

Performance measurement using confusion matrix scheme on 13 labels, namely Data Pendukung, Executive Summary, Informasi Profile Kepesertaan, Kepesertaan Bukan Penerima Upah, Kepesertaan Jasa Konstruksi, Kepesertaan Penerima Upah, Keuangan, Network (Jaringan), Pelayanan, Pengawasan dan Pemeriksaan, Permintaan Antar Kantor (Wilayah), User dan Role, dan e-Channel.. Where measurements will be made on accuracy, precision, and recall. From the validation results carried out by the operator based on the prediction results of the BiLSTM model, 2,822 data were obtained which resulted in wrong predictions. This means that there are 29,178 data or 91.2% of 32,000 data that produce correct prediction results or in accordance with the actual predictions of the operator. From these results, each label is calculated TP, TN, FP and FN to measure the overall performance of Bi-LSTM. The results of the calculation of TP, TN, FP and FN as a whole are shown in Table 2.

Table 2 Confusion Matrix Calculation for each Label

Label	TP	TN	FP	FN	Precision	Recall
-------	----	----	----	----	-----------	--------

Data Pendukung	166	31,568	0	55	1	0.7511
E-Channel	1,671	27,476	921	41	0.6447	0.9761
Executive Summary	10	31,965	0	0	1	1
Informasi Profile	5,456	22,930	0		1	0.7417
Kepesertaan				1,900		
Kepesertaan Bukan	2,154	28,955	0	185	1	0.9209
Penerima Upah						
Kepesertaan Jasa	651	31,273	0	24	1	0.9644
Konstruksi						
Kepesertaan	4,699	25,192	1,901	208	0.7120	0.9576
Penerima Upah						
Keuangan dan	698	31,226	0	4	1	0.9943
Akuntansi						
Network (Jaringan)	2,505	29,231	0	15	1	0.9940
Pelayanan	7,926	23,616	0	340	1	0.9589
Pengawasan dan	411	31,516	0	26	1	0.9405
Pemeriksaan						
Permintaan Antar	383	30,674	0	26	1	0.9364
Kantor (Wilayah)						
User dan Role	2,448	29,448	0	24	1	0.9903
Total	29,178	375,070	2,822	2,848	12.3566	12.1263

The classification process in this study is a multi-label classification so that the calculation process of TP, TN, FP and FN is calculated from each label as well as precision and recall which are also calculated from each label. To measure the overall performance of the Bi-LSTM model, the results of TP, TN, FP, and FN are summed up and then divided by the number of 13 labels. The results of the overall accuracy, all precision and all recall are shown in Table 3

Table 3 Accuracy, All Precision dan All Recall

Accuracy	All Precision	All Recall
0.9118	0.9505	0.9328

From Table 3, obtained the results of accuracy, precision, and recall for 32,000 data testing helpdesk ticket reports, namely accuracy of 91.2%, precision of 95%, and recall of 93% respectively.

5. Conclusion

Based on the test results of the Helpdesk Ticket Classification for Technician Assignment Routes Using BiLSTM, it can be concluded that the test was conducted using 128,000 completed helpdesk ticket report data as training data and 32,000 incomplete helpdesk ticket report data as testing data. The BiLSTM prediction process was tested using the Flask web application that was created and with a one-by-one helpdesk request input scheme. Out of the 32,000 data used as testing data, there were 29,178 data that were validated correctly, or 91.2% of the total data that were predicted correctly or matched the actual prediction from the operator. Overall, the test results showed an accuracy of 91.2%, precision of 95%, and recall of 93%.

References

- [1] H. Kaur, "SENTIMENT ANALYSIS OF USER REVIEW TEXT THROUGH CNN AND LSTM METHODS," p. 17, 2020.
- [2] "T. Wu and G. Zheng, 'Research on Hotel Comment Emotion Analysis Based on BiLSTM and GRU,' 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE), 2021, pp. 143-146, doi: 10.1109/RCAE53607.2021.9638802."

- [3] F. Al-Hawari and H. Barham, "A machine learning based help desk system for IT service management," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 702–718, Jul. 2021, doi: 10.1016/j.jksuci.2019.04.001.
- [4] A. Mandal, N. Malhotra, S. Agarwal, A. Ray, and G. Sridharaf, "Automated Dispatch of Helpdesk Email Tickets: Pushing the Limits with AI," p. 8.
- [5] S. P. Paramesh and K. S. Shreedhara, "IT HELP DESK INCIDENT CLASSIFICATION USING CLASSIFIER ENSEMBLES," *ICTACT JOURNAL ON SOFT COMPUTING*, vol. 09, no. 04, p. 8, 2019.
- [6] R. Baraniuk, D. Donoho, and M. Gavish, "The science of deep learning," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 117, no. 48, pp. 30029–30032, Dec. 2020, doi: 10.1073/pnas.2020596117.
- [7] J. Xiao and Z. Zhou, "Research Progress of RNN Language Model," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China: IEEE, Jun. 2020, pp. 1285–1288. doi: 10.1109/ICAICA50127.2020.9182390.
- [8] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019, doi: 10.1109/ACCESS.2019.2909919.
- [9] P. Hu, J. Tong, J. Wang, Y. Yang, and L. de Oliveira Turci, "A hybrid model based on CNN and Bi-LSTM for urban water demand prediction," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand: IEEE, Jun. 2019, pp. 1088–1094. doi: 10.1109/CEC.2019.8790060.
- [10] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, Aug. 2020, doi: 10.3390/app10175841.
- [11] J. Zheng, "A Novel Computer-Aided Emotion Recognition of Text Method Based on WordEmbedding and Bi-LSTM," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, Dublin, Ireland: IEEE, Oct. 2019, pp. 176–180. doi: 10.1109/AIAM48774.2019.00042.
- [12] Y. Sung, S. Jang, Y.-S. Jeong, and J. H. (James J.) Park, "Malware classification algorithm using advanced Word2vec-based Bi-LSTM for ground control stations," *Computer Communications*, vol. 153, pp. 342–348, Mar. 2020, doi: 10.1016/j.comcom.2020.02.005.
- [13] C. Wang, P. Nulty, and D. Lillis, "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, Seoul Republic of Korea: ACM, Dec. 2020, pp. 37–46. doi: 10.1145/3443279.3443304.
- [14] M. A. Prihandono, R. Harwahyu, and R. F. Sari, "Performance of Machine Learning Algorithms for IT Incident Management," p. 6.
- [15] U. Hasanah, T. Astuti, R. Wahyudi, Z. Rifai, and R. A. Pambudi, "An Experimental Study of Text Preprocessing Techniques for Automatic Short Answer Grading in Indonesian," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia: IEEE, Nov. 2018, pp. 230–234. doi: 10.1109/ICITISEE.2018.8720957.
- [16] J. H. Thrall *et al.*, "Artificial Intelligence and Machine Learning in Radiology: Opportunities, Challenges, Pitfalls, and Criteria for Success," *Journal of the American College of Radiology*, vol. 15, no. 3, pp. 504–508, Mar. 2018, doi: 10.1016/j.jacr.2017.12.026.
- [17] V. B. de Souza, J. C. Nobre, and K. Becker, "Characterization of Anxiety, Depression, and their Comorbidity from Texts of Social Networks," in *Anais do XXXV Simpósio Brasileiro de Banco de Dados (SBBDD 2020)*, Brasil: Sociedade Brasileira de Computação - SBC, Sep. 2020, pp. 121–132. doi: 10.5753/sbbd.2020.13630.
- [18] A. F. de Sousa Neto, B. L. D. Bezerra, A. H. Toselli, and E. B. Lima, "HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Recife/Porto de Galinhas, Brazil: IEEE, Nov. 2020, pp. 54–61. doi: 10.1109/SIBGRAPI51738.2020.00016.
- [19] P. Singla, M. Duhan, and S. Saroha, "An ensemble method to forecast 24-h ahead solar irradiance using wavelet decomposition and BiLSTM deep learning network," *Earth Sci Inform*, vol. 15, no. 1, pp. 291–306, Mar. 2022, doi: 10.1007/s12145-021-00723-1.