

# Multimodal speech emotion recognition optimization using genetic algorithm

Stefanus Michael, Amalia Zahra

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

## Article Info

### Article history:

Received Dec 12, 2023

Revised Feb 15, 2024

Accepted Mar 30, 2024

### Keywords:

A lite bidirectional encoder representation from transformers

Genetic algorithm

Interactive emotional dyadic motion capture

Long short-term memory

Speech emotion recognition

## ABSTRACT

Speech emotion recognition (SER) is a technology that can detect emotions in speech. Various methods have been used in developing SER, such as convolutional neural networks (CNNs), long short-term memory (LSTM), and multilayer perceptron. However, sometimes in addition to model selection, other techniques are still needed to improve SER performance, namely optimization methods. This paper compares manual hyperparameter tuning using grid search (GS) and hyperparameter tuning using genetic algorithm (GA) on the LSTM model to prove the performance increase in the multimodal SER model after optimization. The accuracy, precision, recall, and F1 score improvement obtained by hyperparameter tuning using GA (HTGA) is 2.83%, 0.02, 0.05, and 0.04, respectively. Thus, HTGA obtains better results than the baseline hyperparameter tuning method using a GS.

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



## Corresponding Author:

Stefanus Michael

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

Bina Nusantara University

St. Kebon Jeruk Number 27, Kebon Jeruk, Jakarta Barat, Jakarta, Indonesia

Email: stefanus.michael@binus.ac.id

## 1. INTRODUCTION

Speech emotion recognition (SER) is a field of research that explores the detection of emotions from human speech. This speech can come from sound recordings made for research purposes or the result of human interaction with technology. The application of SER technology can be done in various fields, such as customer service [1], military [2], and transportation systems [3]. In addition, SER is also used in online learning [4] and health care [5]. The application of SER technology in these different fields can provide numerous benefits, such as improving the quality of customer service, improving user experience in the transportation system, refining the quality of online learning, and assisting the screening process in health care.

An extraction process must occur before an audio file can be further processed to construct an SER. The audio file extraction produces features such as zero-crossing rate, linear predictor coefficient (LPC), and mel-frequency cepstral coefficient (MFCC) [6]. In other SER studies [7], the feature extraction process uses a feature set, which extracts a specific series of features using particular libraries or tools. In the research [7], the feature set used is geneva minimalistic acoustic parameter set (GeMAPS). After feature extraction, selecting a classifier algorithm that will be used as the basis of the SER model is necessary. Several algorithms such as convolutional neural network (CNN) [8], long short-term memory (LSTM) [9], support vector machine (SVM) [10], and bidirection recurrent unit (BGRU) [11] are used as classifiers in SER research. In addition to these SER-related studies, several other studies have used optimization algorithms to improve SER performance. Some optimization algorithms that have been used in the development of SER are quantum-behaved particle swarm optimization (QPSO) [12], which uses the modified QPSO optimization

method to obtain an optimal dimension reduction matrix, and genetic algorithm (GA) [13] that used clustering based GA to optimize features.

Apart from speech, text is also known as one of the data sources that can be used for emotion detection. Features must be extracted from text data to develop emotion detection in text. Term frequency (TF) [14], term frequency-inverse document frequency (TFIDF) [14], [15], and bag-of-words (BOW) [16] are examples of features that have been used in emotion detection in text. Besides that, there are several studies involving pre-trained models to develop emotion detectors in text, such as bidirectional encoder representations from transformers (BERT) [17] and a lite BERT (ALBERT) [18]. In addition, several studies [17]-[21] add text data as an additional source to develop SER to increase detection accuracy. SER research involving text data is referred to as bimodal or multimodal SER.

Research that focuses on optimizing the SER model by performing hyperparameter tuning on the multimodal SER model with voice and text data sources has not been widely explored. One that can be found is [22], which uses the Bayesian and random forest methods. Thus, we are interested in exploring this area of research further. On the other hand, GA has been successfully used to optimize models on various topics, such as prediction of sepsis [23] and spam prediction [24], and as far as the authors know, GA has never been applied to perform hyperparameter tuning on multimodal SER model with interactive emotional dyadic motion capture (IEMOCAP) dataset, so it is not yet known how much performance improvement results from using GA for optimization in this case.

The contribution of this paper is to find out how much optimization using the GA algorithm can improve the performance of an LSTM-based SER system so that other researchers can obtain more detailed consideration in choosing optimization methods for similar research. Optimization needs to be done because it is one way that can be used to improve model performance. Optimization is done by performing hyperparameter tuning of the LSTM model using the GA algorithm. The LSTM model was chosen because it is a model from the SER research using the IEMOCAP dataset with the best results as far as the author knows [9]. While the GA algorithm was chosen because it has superior performance compared to several other optimization algorithms, such as particle swarm optimization (PSO) [25], as well as the bayesian algorithm (BA) and grid search (GS) [26]. In addition, we were inspired by the LSTM model optimization experiment using GA conducted by [27]. We are interested in applying a similar optimization method in this study. Because as far as we know, this method has never been applied to multimodal SER cases. Then, to measure the performance improvement obtained, hyperparameter tuning using GS (HTGS) will be used as a baseline because it is commonly used to perform hyperparameter tuning in other similar studies [28]-[30].

The structure of this paper is as follows: section 1 provides an introduction to the topic of this research. Section 2 describes the method and provides an overview of the experiment in detail. Section 3 describes the experiment's results. Finally, section 4 describes the conclusions obtained.

## 2. METHOD

This research uses the IEMOCAP dataset [31]. This dataset was chosen because it is commonly used in similar studies [7], [9], [18], [21], [32]-[36]. The IEMOCAP dataset contains voice data samples, text transcription, video, and motion capture from faces for the purposes of emotion detection research. However, the data used in this study are only voice and text. The actors involved in the dataset creation were ten people, five men and five women, with 10,039 voice data pieces produced with an average duration of 4.5 seconds. The dataset is divided into nine categories of emotions, namely excited, frustrated, happiness, neutral, anger, surprise, sadness, disgust, and fear. However, the emotion classes used in this study are limited to four: angry with a total sample of 1,103 samples, excited 1,041 samples, neutral 1,708 samples, and sadness 1,084 samples. We chose to use four emotions because similar studies have been done before using these four emotional classes [33], [35], [36]. Specifically for text data only, pre-processing is carried out in the form of lowercase, removing stop words, removing punctuation, removing numbers, and removing lemmatization. The structure of the research implementation can be seen in Figure 1.

The research begins by extracting features from the IEMOCAP dataset using the openSMILE library [37] for speech features and ALBERT [38] for text features. The configuration used to extract speech features is eGeMAPS [39], which contains 88 feature parameters. The features included in this feature set are MFCC, pitch, jitter, and others. For more detailed information about its features, see the reference for eGeMAPS [39]. Meanwhile, the "albert-base-v2" configuration is used to extract text features.

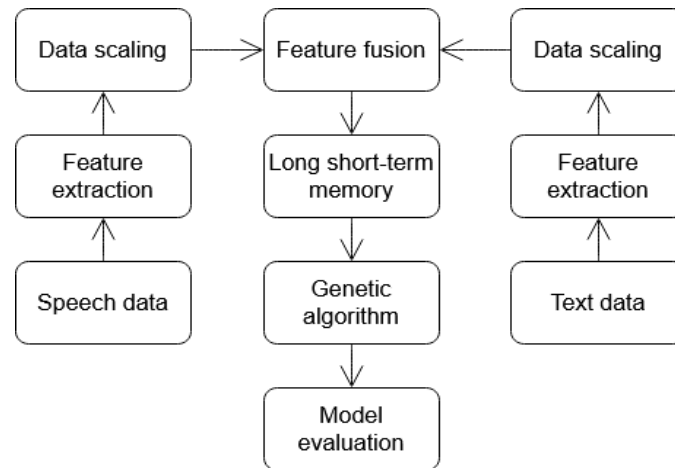


Figure 1. Structure of research

Feature extraction will produce text features with dimensions (4,936, 64, 768) where 4,936 is the number of samples, 64 is the vocabulary size, and 768 is the hidden layer size. Meanwhile, the sound features produced have dimensions (4,936, 88), with 4,936 being the number of samples and 88 being the number of feature parameters produced. After that, the data extraction results will go through a data scaling process using MinMaxScaler from sklearn library [40]. Features that have gone through data scaling will be combined directly by first padding the speech data using the feature speech value. The padding process begins with expanding the dimensions using the numpy library [41] so that the sound feature has dimensions (4,936, 1, 88). Then padding is done on the y-axis, with the dimension axis expressed as (x, y, z), as many as 63 rows using the same values as those on the z-axis of the speech feature. Using the same value aims to avoid decreasing the effect of speech features when combined with text features compared to using a zero value for padding. So, the sound feature after padding has dimensions (4,936, 64, 88). After having the exact dimensions on the x-axis and y-axis, the speech features are combined with the text features based on the z-axis so that the combined features have dimensions (4,936, 64, 856), where 4,936 is the number of samples, 64 is the vocabulary size of the text feature, and 856 is the number of combined columns between feature speech and text.

After that, the LSTM model will be formed with a total of seven layers, consisting of the first layer is an input layer, the second layer is an LSTM layer with neurons amount equal to values being tested (obtained from the GS process or GA), the third layer is a dropout layer with a rate equal to the tested value, the fourth layer is an LSTM layer with neurons amount equals to half the value being tested, the fifth layer is a fully connected layer with a total of 128 neurons and a relu activation function, the sixth layer is a dropout layer with a rate equal to the tested value, and the last layer is a fully connected layer as an output layer with a number of neurons equals to 4 and a SoftMax activation function. Then, we train the LSTM model using the multimodal feature with two scenarios, namely HTGS and hyperparameter tuning using GA (HTGA). HTGS will be carried out by testing hyperparameter values one by one using a regular GS algorithm without any modifications as the baseline model. In contrast, HTGA will be carried out by making a series of hyperparameter values as chromosomes in GA and testing hyperparameter values using the GA algorithm. The pseudocode of the GA algorithm used in this research can be seen in Figure 2. The hyperparameter tuning process will use data validation as data for model evaluation. Then, the best model will be re-evaluated using test data for the final result. We compare the results of the two experiments in section 3.

```

best chromosome: list
best chromosome performance: float
children: list
crossover rate: float
mutation rate: float
population: list
performance: list //performance of chromosome in population list
search: list //contains all value in experiment range value for GA experiment
selected parent: list
selection: integer
selected: list //selected chromosome for parent in crossover process

Begin
  create initial population
  best chromosome = 0
  best chromosome performance = 0
  While (a < generation length)
    for i=1 to population amount
      Calculate performance score of chromosome i
    end for
    // get best chromosome and best performance score from current population
    for i=1 to population length
      if best chromosome performance < performance of chromosome[i]
        best chromosome = chromosome[i]
        best chromosome performance = performance of chromosome [i]
      end for
    // tournament system for parent selection
    for i=1 to population length
      selection = get random integer (between 1 and population length)
      x = get two random integer (between 1 and population length)
      for i=1 to 2
        if performance[selection] < performance[x[i]]
          selection = x[i]
        end for
      add selection to selected list
    end for
    // crossover process
    for i=1 to population length with step 2
      crossover point = get random integer (between 1 and gene length-1)
      parent 1 = selected[i]
      parent 2 = selected[i+1]
      child 1 = get gene 1 to gene crossover point from parent 1 + get gene after crossover point to gene 4 from parent 2
      child 2 = get gene 1 to gene crossover point from parent 2 + get gene after crossover point to gene 4 from parent 1
      add child 1 and child 2 to children list
    end for
    // mutation process
    for i=1 to children length
      for j=1 to gene length
        mutate = get random float (between 0 and 1)
        if mutate < mutation rate
          gene mutate = get random integer (between gene experiment range value length)
          children[i][j] = search[j][gene mutate]
        end if
      end for
    end for

    // children become the next population
    population = children
  end while
End

```

Figure 2. Pseudocode of HTGA experiment

### 3. RESULTS AND DISCUSSION

The experiment was carried out following the structure of the research implementation discussed in section 3. The data used is divided into train, validation, and test data with a percentage of 70%, 15%, and 15%, respectively. (HTGS) uses experiment range values: 10 to 130 with step 60 for batch size, 0.1 to 0.5 with step 0.2 for dropout, 100 to 800 with step 350 for LSTM neuron, and 0.0001 to 0.005 with step 0.0025 for learning rate. HTGS produces a combination of hyperparameters: 130 for batch size, 0.3 for dropout, 450 for LSTM neurons, and 0.0001 for learning rate. The experiment data for (HTGS) can be seen in Table 1, while the experiment data for HTGA can be seen in Table 2.

Table 1. HTGS

Hyperparameter	Value range	Step	Result value
Batch size	10-130	60	130
Dropout	0.1-0.5	0.2	0.3
LSTM neuron	100-800	350	450
Optimizer learning rate	0.0001-0.005	0.0025	0.0001

Then, the best model from HTGS is evaluated using test data. The HTGS experiment obtained an accuracy of 66.67%, precision of 0.68, recall of 0.65, and F1 score of 0.66. Figure 3 shows that the number of samples correctly predicted per emotion for HTGS is 113 for angry, 63 for excited, 194 for neutral, and 124 for sadness. The results of the highest number of correct predictions are in the neutral emotion class. This can happen because the emotion class has the largest number of samples in this study, thus providing more variety to be learned by the SER model used. Likewise, the prediction results of the angry and sadness emotion classes which have a slightly different number of samples, have the correct prediction results with similar numbers.

Table 2. HTGA

Hyperparameter	Value range	Step	Result value
Batch size	10-130	5	30
Dropout	0.1-0.5	0.05	0.25
LSTM neuron	100-800	10	290
Optimizer learning rate	0.0001-0.005	0.0001	0.0002

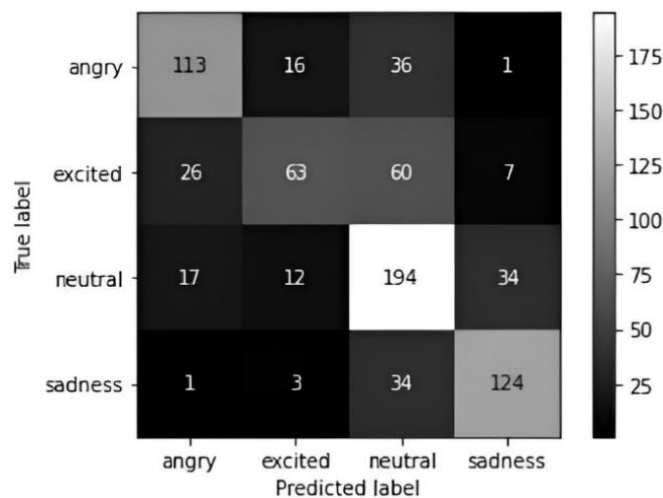


Figure 3. Confusion matrix from the result of HTGS

Whereas in the excited emotion class, although the number of samples is not much different from the angry and sadness emotion classes, the number of correct predictions is quite different from the two classes. This can happen because of the similarity of the characteristics of the emotional classification of the excited class with those of the two classes, such as the sometimes high-pitched voice that makes it look like an angry emotion or the choice of words (in the text) that are similar to neutral emotions (e.g. "OK, let's go!" can be read in an excited tone or a neutral tone). Thus, excited emotion samples are misclassified as angry and neutral emotions. Meanwhile, for most misclassifications, apart from excited emotions, which are classified as neutral, there are also neutral emotions, incorrectly classified as sadness, and vice versa. One of the things that allows this to happen is the similarity in the typical flat tone characteristic of the two emotions.

After the manual hyperparameter tuning experiment, we performed hyperparameter tuning of the LSTM model using the GA algorithm. Hyperparameter tuning with GA uses range experiment values as follows: 10 to 130 with step 5 for batch size, 0.1 to 0.5 with step 0.05 for dropout, 100 to 800 with step 10 for LSTM neuron, 0.0001 to 0.005 with step 0.0001 for learning rate. HTGA produces a combination of hyperparameters: 30 for batch size, 0.25 for dropout, 290 for LSTM neuron, and 0.0002 for learning rate, which can be seen in Table 2. The steps in HTGA are different from those in HTGS because, in HTGA, hyperparameter tuning is done automatically using GA, thus making it possible to have more value variations. Then, the best model from HTGA is evaluated using test data and produces an accuracy of 69.50%, precision of 0.70, recall of 0.70, and F1 score of 0.70. The comparison of manual HTGS and HTGA results can be seen in Table 3. Figure 4 shows that the number of samples correctly predicted per emotion for HTGA is 119 for angry, 100 for excited, 176 for neutral, and 120 for sadness. The analysis of the results of the number of correct predictions from the HTGA is similar to that for the HTGS.

Table 3. Comparison of HTGS and hyperparameter using GA result

Experiment	Accuracy (%)	Precision	Recall	F1 score
HTGS	66.67	0.68	0.65	0.66
HTGA	69.50	0.70	0.70	0.70

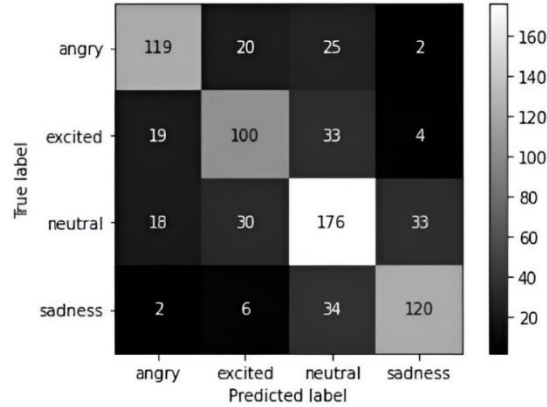


Figure 4. Confusion matrix from the result of HTGA

HTGA can obtain better results than HTGS due to more testing variations of hyperparameter values. Thus, the limitations in HTGS, such as the number of variations of hyperparameter values tested for each hyperparameter, can be resolved by performing HTGA. The number of variations limitation in HTGS exists because HTGS requires testing each combination of hyperparameter values, making applying more variations of hyperparameter values impractical as it requires too much time and resources.

#### 4. CONCLUSION

The SER experiment using the LSTM model with hyperparameter tuning with GA successfully resulted in a reasonably good performance improvement. The accuracy obtained after manual hyperparameter tuning is 66.67% and after hyperparameter tuning with GA is 69.50%. It can be seen that the hyperparameter tuning process, which is carried out automatically by utilizing the GA algorithm, can improve the accuracy performance of the SER model by 2.83% in this experiment. For future research potential, SER model ensembles can be carried out by adding input such as video.

#### ACKNOWLEDGEMENTS





The author thanks Binus University for supporting this research.

#### REFERENCES





- [1] W. Han, T. Jiang, Y. Li, B. Schuller, and H. Ruan, "Ordinal learning for emotion recognition in customer service calls," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6494–6498, 2020, doi: 10.1109/ICASSP40776.2020.9053648.
- [2] S. Tokuno *et al.*, "Usage of emotion recognition in military health care," *2011 Defense Science Research Conference and Expo (DSR)*, Singapore, 2011, pp. 1–5, doi: 10.1109/DSR.2011.6026823.
- [3] L. Tan *et al.*, "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space-air-ground integrated intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2830–2842, 2022, doi: 10.1109/TITS.2021.3119921.
- [4] Y. Jiang and X. Li, "Intelligent online education system based on speech recognition with specialized analysis on quality of service," *International Journal of Speech Technology*, vol. 23, no. 3, pp. 489–497, 2020, doi: 10.1007/s10772-020-09723-w.
- [5] M. S. Hossain and G. Muhammad, "An audio-visual emotion recognition system using deep learning fusion for a cognitive wireless framework," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 62–68, 2019, doi: 10.1109/MWC.2019.1800419.
- [6] P. Jiang, H. Fu, H. Tao, P. Lei, and L. Zhao, "Parallelized convolutional recurrent neural network with spectral features for speech emotion recognition," *IEEE Access*, vol. 7, pp. 90368–90377, 2019, doi: 10.1109/ACCESS.2019.2927384.
- [7] B. T. Atmaja and M. Akagi, "Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning," *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e17, 2020, doi: 10.1017/ATSIP.2020.14.
- [8] D. N. Krishna and A. Patil, "Multimodal emotion recognition using cross-modal attention and 1D convolutional neural networks," in *INTERSPEECH 2020*, 2020, pp. 4243–4247, doi: 10.21437/Interspeech.2020-1190.
- [9] G. Shen *et al.*, "WISE: Word-level interaction-based multimodal fusion for speech emotion recognition," in *INTERSPEECH*

- 2020, 2020, pp. 369–373, doi: 10.21437/Interspeech.2020-3131.
- [10] J. Ancilin and A. Milton, “Improved speech emotion recognition with mel frequency magnitude coefficient,” *Applied Acoustics*, vol. 179, p. 108046, 2021, doi: 10.1016/j.apacoust.2021.108046.
- [11] Z. Zhu, W. Dai, Y. Hu, and J. Li, “Speech emotion recognition model based on Bi-GRU and Focal Loss,” *Pattern Recognition Letters*, vol. 140, pp. 358–365, 2020, doi: 10.1016/j.patrec.2020.11.009.
- [12] F. Daneshfar and S. J. Kabudian, “Speech emotion recognition using discriminative dimension reduction by employing a modified quantum-behaved particle swarm optimization algorithm,” *Multimedia Tools and Applications*, vol. 79, no. 1–2, pp. 1261–1289, 2020, doi: 10.1007/s11042-019-08222-8.
- [13] S. Kanwal and S. Asghar, “Speech emotion recognition using clustering based GA-optimized feature set,” *IEEE Access*, vol. 9, pp. 125830–125842, 2021, doi: 10.1109/ACCESS.2021.3111659.
- [14] A. Yousaf *et al.*, “Emotion recognition by textual tweets classification using voting classifier (LR-SGD),” *IEEE Access*, vol. 9, pp. 6286–6295, 2021, doi: 10.1109/ACCESS.2020.3047831.
- [15] E. Batbaatar, M. Li, and K. H. Ryu, “Semantic-emotion neural network for emotion recognition from text,” *IEEE Access*, vol. 7, pp. 111866–111878, 2019, doi: 10.1109/ACCESS.2019.2934529.
- [16] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, “Deep learning for affective computing: Text-based emotion recognition in decision support,” *Decision Support Systems*, vol. 115, pp. 24–35, 2018, doi: 10.1016/j.dss.2018.09.002.
- [17] N. H. Ho, H. J. Yang, S. H. Kim, and G. Lee, “Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network,” *IEEE Access*, vol. 8, pp. 61672–61686, 2020, doi: 10.1109/ACCESS.2020.2984368.
- [18] M. Chen and X. Zhao, “A multi-scale fusion framework for bimodal speech emotion recognition,” in *INTERSPEECH 2020*, 2020, pp. 374–378, doi: 10.21437/Interspeech.2020-3156.
- [19] S. Yoon, S. Byun, S. Dey, and K. Jung, “Speech emotion recognition using multi-hop attention mechanism,” *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2822–2826, 2019, doi: 10.1109/ICASSP.2019.8683483.
- [20] P. Singh, R. Srivastava, K. P. S. Rana, and V. Kumar, “A multimodal hierarchical approach to speech emotion recognition from audio and text,” *Knowledge-Based Systems*, vol. 229, p. 107316, 2021, doi: 10.1016/j.knosys.2021.107316.
- [21] B. T. Atmaja and M. Akagi, “Two-stage dimensional emotion recognition by fusing predictions of acoustic and text networks using SVM,” *Speech Communication*, vol. 126, pp. 9–21, 2021, doi: 10.1016/j.specom.2020.11.003.
- [22] A. B. Gumelar, E. M. Yuniarno, D. P. Adi, A. G. Soai, I. Sugiarto, and M. H. Purnomo, “BiLSTM-CNN hyperparameter optimization for speech emotion and stress recognition,” in *2021 International Electronics Symposium (IES)*, Surabaya, Indonesia, 2021, pp. 156–161, doi: 10.1109/IES53407.2021.9594024.
- [23] P. Nejedly, F. Plesinger, I. Viscor, J. Halamek, and P. Jurak, “Prediction of sepsis using LSTM with hyperparameter optimization with a genetic algorithm,” in *2019 Computing in Cardiology*, vol. 45, pp. 2–5, 2019, doi: 10.22489/cinc.2019.022.
- [24] N. Ghatasheh, I. Altaharwa, and K. Aldebei, “Modified genetic algorithm for feature selection and hyper parameter optimization: Case of XGBoost in spam prediction,” *IEEE Access*, vol. 10, no. July, pp. 84365–84383, 2022, doi: 10.1109/ACCESS.2022.3196905.
- [25] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, “Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting,” *Energies*, vol. 13, no. 391, pp. 1–21, 2020, doi: 10.3390/en13020391.
- [26] H. Alibrahim and S. A. Ludwig, “Hyperparameter optimization: comparing genetic algorithm against grid search and Bayesian optimization,” in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, doi: 10.1109/CEC45853.2021.9504761.
- [27] N. Gorgolis, I. Hatzilygeroudis, Z. Istenes, and L. G. Gyenne, “Hyperparameter optimization of LSTM network models through genetic algorithm,” in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2019, pp. 1–4, doi: 10.1109/IISA.2019.8900675.
- [28] H. N. Zahra, M. O. Ibrohim, J. Fahmi, R. Adelia, F. A. N. Febryanto, and O. Riandi, “Speech emotion recognition on Indonesian youtube web series using deep learning approach,” *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–6, doi: 10.1109/ICIC50835.2020.9288650.
- [29] R. Marin and D. Valles, “A speech emotion recognition solution-based on support vector machine for children with autism spectrum disorder to help identify human emotions,” *2020 Intermountain Engineering, Technology and Computing (IETC)*, 2020, pp. 1–6, doi: 10.1109/IETC47856.2020.9249147.
- [30] S. Sultana and M. S. Rahman, “Acoustic feature analysis and optimization for Bangla speech emotion recognition,” *Acoust. Sci. Tech.*, vol. 44, no. 3, pp. 157–166, 2023, doi: 10.1250/ast.44.157.
- [31] C. Busso *et al.*, “IEMOCAP: Interactive emotional dyadic motion capture database,” *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008, doi: 10.1007/s10579-008-9076-6.
- [32] B. T. Atmaja, R. Elbarougy, and M. Akagi, “Dimensional speech emotion recognition from acoustic and text features using recurrent neural networks,” *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM)*, vol. 1, no. 1, pp. 91–102, 2020, doi: 10.34010/injiiscom.v1i1.4023.
- [33] L. Cai, Y. Hu, J. Dong, and S. Zhou, “Audio-textual emotion recognition based on improved neural networks,” *Mathematical Problems in Engineering*, vol. 2019, pp. 1–9, 2019, doi: 10.1155/2019/2593036.
- [34] J. Sebastian and P. Pierucci, “Fusion techniques for utterance-level emotion recognition combining speech and transcripts,” *Interspeech 2019*, 2019, pp. 51–55, doi: 10.21437/Interspeech.2019-3201.
- [35] M. Xu, F. Zhang, and W. Zhang, “Head fusion: improving the accuracy and robustness of speech emotion recognition on the IEMOCAP and RAVDESS dataset,” *IEEE Access*, vol. 9, pp. 74539–74549, 2021, doi: 10.1109/ACCESS.2021.3067460.
- [36] M. S. Fahad, R. Raj, A. Ranjan, and A. Deepak, “Discriminative feature construction using multi-labeling approach for automatic speech emotion recognition,” in *Machine Intelligence Techniques for Data Analysis and Signal Processing*, 2023, pp. 869–880, doi: [https://doi.org/10.1007/978-981-99-0085-5\\_71](https://doi.org/10.1007/978-981-99-0085-5_71).
- [37] F. Eyben, M. Wöllmer, and B. Schuller, “OpenSMILE - The Munich versatile and fast open-source audio feature extractor,” in *Proceedings of the 18th ACM international conference on Multimedia (MM '10)*, 2010, pp. 1459–1462, doi: 10.1145/1873951.1874246.
- [38] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” *arXiv preprint arXiv:1909.11942*, 2019.
- [39] F. Eyben *et al.*, “The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing,” *IEEE Transactions on Affective Computin.*, vol. 7, no. 2, pp. 190–202, 2016, doi: 10.1109/TAFFC.2015.2457417.
- [40] F. Pedregosa *et al.*, “Scikit-learn: machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [41] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

**BIOGRAPHIES OF AUTHORS**

**Stefanus Michael**     is a student in Bina Nusantara University. His research interests include speech emotion recognition and image processing. He can be contacted at email: stefanus.michael@binus.ac.id.



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