Published online on the journal's webpage: **http://journal.ittelkom-pwt.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 4 No. 2 (2024) 63 - 74 E-ISSN: 2809-8064

CNN-LSTM for MFCC-based Speech Recognition on Smart Mirrors for Edge Computing Command

Aji Gautama Putrada[1*], Ikke Dian Oktaviani[2], Mohamad Nurkamal Fauzan[3], Nur Alamsyah[4]

[1,2,3,4*]School of Computing, Telkom University

[1*]ajigps@telkomuniversity.ac.id, [2]oktavianiid@telkomuniversity.ac.id, [3]mnurkamalfauzan@student.telkomuniversity.ac.id, [4]nuralamsyah@student.telkomuniversity.ac.id

Abstract

Smart mirrors are conventional mirrors that are augmented with embedded system capabilities to provide comfort and sophistication for users, including introducing the speech command function. However, existing research still applies the Google Speech API, which utilizes the cloud and provides sub-optimal processing time. Our research aim is to design speech recognition using Mel-frequency cepstral coefficients (MFCC) and convolutional neural network–long short-term memory (CNN-LSTM) to be applied to smart mirror edge devices for optimum processing time. Our first step was to download a synthetic speech recognition dataset consisting of waveform audio files (WAVs) from Kaggle, which included the utterances “left,” “right,” “yes,” “no,” “on,” and “off. ” We then designed speech recognition by involving Fourier transformation and low-pass filtering. We benchmark MFCC with linear predictive coding (LPC) because both are feature extraction methods on speech datasets. Then, we benchmarked CNN-LSTM with LSTM, simple recurrent neural network (RNN), and gated recurrent unit (GRU). Finally, we designed a smart mirror system complete with GUI and functions. The test results show that CNN-LSTM performs better than the three other methods with accuracy, precision, recall, and an f1-score of 0.92. The speech command with the best precision is "no," with a value of 0.940. Meanwhile, the command with the best recall is "off," with a value of 0.963. On the other hand, the speech command with the worst precision and recall is "other," with a value of 0.839. The contribution of this research is a smart mirror whose speech commands are carried out on the edge device with CNN-LSTM.

Keywords: *convolutional neural network-long short-term memory, Mel-frequency cepstral coefficients, smart mirror, speech recognition, edge computing*

1. Introduction

Smart mirrors are conventional mirrors augmented with embedded system capabilities to provide comfort and sophistication for users [1]. Until now, progress in smart mirror research has reached several stages, for example Bianco *et al.* [2] created a smart mirror that can detect emotions, which can be useful for content recommendations. Majumder *et al.* [3] created a smart mirror with features such as date and time, weather, and news updates, which can be realized with several APIs installed on the Raspberry Pi. Tater *et al.* [4] implemented all the features from the previous mentioned research then they added new features, namely maps, gestures, and reminders.

On the other hand, research has implemented speech or voice commands on smart mirrors, some with cloud support. Yui *et al.* [5] created a voice command on a smart mirror with a cloud library called Sonus, which can carry out processing to capture hotwords. Shakir *et al.* [6] created a smart mirror with various capabilities where voice commands are carried out via smartphone and connected to other devices at home. On the other hand, studies have proven that cloud computing increases processing time in real-time systems [7]. Applying voice commands directly to smart mirror devices with the edge computing concept can be a research opportunity for optimum processing time.

Several studies use the long short-term memory (LSTM) model as pattern classification in speech recognition. Jo *et al.* [8] tried to make the LSTM model efficient because they wanted it implemented on a low-resource computer. The paper makes it efficient by matching similar LSTM neurons and diminishing them. Other studies such as Oruh *et al.* [9] combines LSTM with other models to improve LSTM performance in speech

Received: 07-06-2024 | Accepted: 27-06-2024 | Published: 01-08-2024

recognition. This research improves performance by adding a recurrent neural network (RNN) layer before LSTM to increase memory capacity in the forget gate. There has never been any research that applies LSTM speech recognition to smart mirrors.

Mel-frequency cepstral coefficients (MFCCs) are feature extraction methods in speech recognition whose output is a two-dimensional structure [10]. On the other hand, two-dimensional-convolutional neural network (2D-CNN) produces feature maps from two-dimensional structural data such as images which are useful for deep learning [11]. Several studies have utilized CNN-LSTM to improve speech recognition performance, such as that done by Alsayadi *et al.* [12] in Arabic. Applying CNN-LSTM to the MFCC map and testing its performance is a research opportunity.

Our research aim is to design speech recognition using MFCC and CNN-LSTM to be applied to smart mirrors with an edge computing concept that boosts processing time efficiency. Our first step was downloading a synthetic speech recognition dataset consisting of waveform audio files (WAVs) from Kaggle. We then designed speech recognition using a flow that utilizes Fourier transformation, low-pass filtering, the MFCC, and CNN-LSTM prediction. We benchmark MFCC with linear predictive coding (LPC) because both are feature extraction methods on speech datasets. We then benchmark the CNN-LSTM with LSTM, simple RNN, and gated recurrent unit (GRU), all of which are recurrent deep learning methods. Finally, we designed a smart mirror system complete with GUI and functions.

To the best of our knowledge, no one has ever used CNN-LSTM and MFCC-based speech recognition for voice commands on smart mirrors. The following is a list of our contributions:

1) MDI as a feature comparison method between MFCC and LPC feature extraction.
2) A CNN-LSTM model for speech recognition with a case study of smart mirror speech command, where CNN-LSMT is the most optimal model compared to other methods.
3) A smart mirror with an edge computing concept for speech command recognition.

The remainder of this paper is written systematically: Section 2 discusses our research methodology. In Section 3 we have carried out testing and the results are presented. In this section, we also compare our test results with state-of-the-art research and formulate research contributions. Section 4 contains answers to our research objectives.

2. Research Methods

We created a methodology to achieve our research aim. The dataset for synthetic speech recognition, consisting of WAV recordings, was downloaded from Kaggle. Next, we create a speech recognition pipeline that uses CNN-LSTM, MFCC, low-pass filtering, and Fourier transform. Since both are feature extraction techniques on voice datasets, we compare MFCC and LPC. We compare CNN-LSTM with three iterative deep-learning techniques: LSTM, basic RNN, and GRU. Ultimately, we created a smart mirror system with a graphical user interface and features. Figure 1 explains our research workflow in block diagram form.

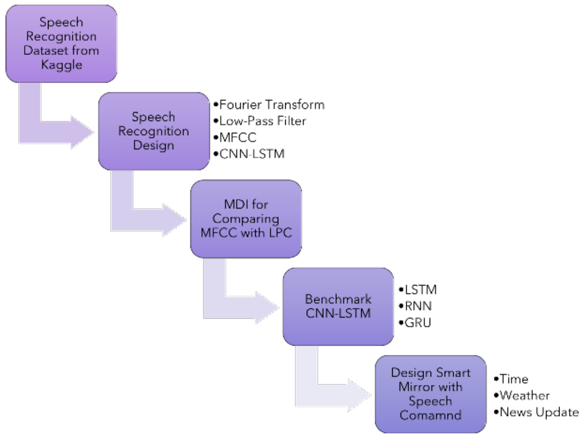


Figure 1. The research workflow.

2.1 *Edge Computing-Based Smart Mirror Design*

A smart mirror is an advanced interactive device with a reflective surface that provides useful personalized information for users [13]. These functions are usually supplemented by sophisticated interaction methods such as touch screens and speech recognition. The three main deployment targets for smart mirrors are housing, healthcare, and retail stores [14]. In housing, smart mirrors become an integral part of smart homes because of their speech recognition capabilities [15]. In healthcare, smart mirrors can display monitoring results from vital signs [16]. Finally, in retail stores, smart mirrors can offer virtual try-ons so that users can see what clothes look like on them without needing to change [17].

The devices involved in the smart mirror to perform the tasks discussed, including speech recognition, are digital displays (LCD TV), Raspberry Pi, and microphones [18]. Then, some APIs are useful for retrieving personalized information, such as weather and news updates from the cloud [19]. Other equipment needed is a two-way mirror to cover the digital display and display the reflection of the user and speakers for interactive response. Raspberry Pi is also equipped with communication protocols such as Wi-Fi for interaction with the cloud. Figure 2 shows the design of our proposed smart mirror system.

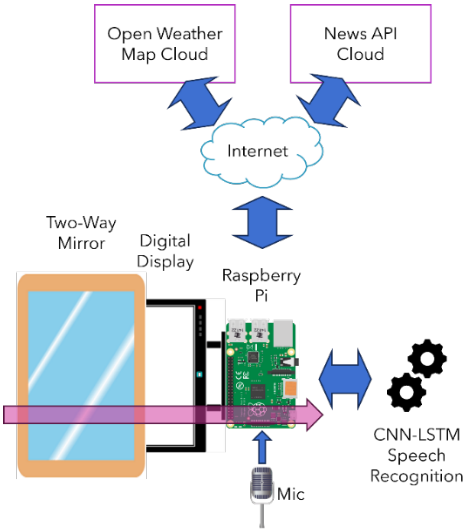


Figure 2. The proposed architecture for edge computing-based smart mirror which includes the CNN-LSTM speech recognition.

Furthermore, we use Flask to fetch weather and news update data from the Open Weather Map Cloud and News API Cloud. Flask is a lightweight web framework from Python that can create servers and manage request and response functions for certain services [20]. By setting up a Flask route, we can define endpoints that communicate with these APIs, receive the desired weather data and news updates, and then display them on the smart mirror. Flask can also make periodic requests and responses so that the information on the smart mirror remains in real-time [21].

Moreover, edge computing means moving part of the processing in an IoT system from the cloud server to a processing unit located closer to the end device or within the end device itself [22]. In the case of this research, the end device is a smart mirror. This means that the Raspberry Pi, acting as the end device in the smart mirror, runs selective smart mirror computations. In this research, we opt to run speech recognition in the smart mirror. We embed speech recognition with Python programming in the Raspberry Pi, which. leverages CNN-LSTM.

2.2 *Speech Recognition with MFCC and CNN-LSTM*

Speech recognition starts from a WAV-formed voice dataset and goes through several stages. These stages are pre-processing, digital signal processing (DSP), MFCC feature selection, and CNN-LSTM training and evaluation. Pre-processing is a series of alterations applied to raw data so that the data is ready to be handled in machine learning. Meanwhile, DSP involves mathematical and programming methods to understand and convert signals such as audio, video, and sensor signals. Figure 3 shows all the stages in the form of a flow chart.

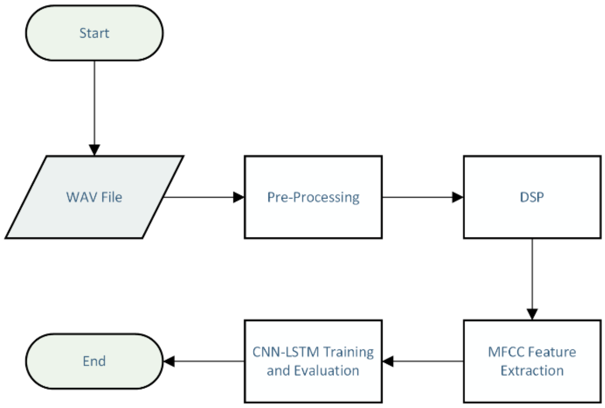


Figure 3. The process of speech recognition.

WAV is a widely used audio file which is a raw and uncompressed audio data format, so it has rich sound information [23]. WAV files developed by IBM and Microsoft consist of information such as audio data and format specifications [24]. To save format specifications, a WAV file has an encoding system that manages to store various kinds of information other than voice data [25]. Information regarding the number of channels is 2 bytes and is in bytes 22 to 23. Next, information about the frame rate is 4 bytes, located in bytes numbers 24 to 27. Then, information in relation to frame size is 4 bytes long; its location is in bytes 28 to 31. Finally, information with respect to sample width is 2 bytes long, located in bytes 34 to 35.

Pre-processing consists of three sub-stages: WAV to Numpy array conversion, channel separation, and normalization. Numpy array conversion converts a bit stream into a logical array with a certain data type, we use a 16-bit integer data type. These changes will simplify high-level operations on the data. Because the WAV channel type has been obtained in decoding the WAV file in the previous step, in this step, if the channel is stereo, then channel separation is carried out. The last step is normalization, where this stage is important to ensure that the amplitude range throughout the dataset has consistent values [26]. The normalization formula is as follows:

$$x' = \frac{x}{|X|_{max}}, x \in X \tag{1}$$

where X is the dataset and x' is the normalized result of x.

The next step in speech recognition is DSP, where this step consists of two sub-steps: Fourier transform and low-pass filtering. Fourier transform converts signals from the time domain to the frequency domain [27]. Fourier transform is useful in speech recognition for frequency analysis and is also useful for the MFCC feature selection stage. The following is the formula for discrete Fourier transform (DFT):

$$X'[k] = \sum_{n=0}^{N-1} X[n] \cdot e^{-j\frac{2\pi}{N}kn} \tag{2}$$

where $X'[k]$ is the output DFT, N is the amount of data analyzed, X is the time domain signal input, and j is an imaginary number. A low-pass filter removes high-frequency components in the signal and removes noise [28]. The function of the low-pass filter in speech recognition is to improve the performance of speech features in the dataset. The following is the formula for a low-pass filter:

$$X'[n] = X[n] * H[n] = \sum_{m=-\infty}^{\infty} X[m] \cdot H[n-m] \tag{3}$$

where X' is the result of the low-pass filter, $*$ is the convolution operation, and H is the impulse response of the filter.

The step after DSP in speech recognition is MFCC feature selection. MFCC is a process that captures the short-term power spectrum of a sound signal [29]. The meaning of MFCC is a cepstral representation of an audio clip by performing a Fourier transform on a signal window and mapping its power on the Mel scale. This process's results resemble the human ear's sensitivity to different frequencies. The MFCC ($M(n)$) process involves the discrete cosine transform (DCT) of the log filterbank energy, where the formula is as follows:

$$M(n) = \sum_{k=1}^{K} log(S(k))cos\left[\frac{\pi n}{K}\left(k-\frac{1}{2}\right)\right], n \in N \tag{4}$$

where $S(k)$ is the Mel scale, K is the number of Mel scales, and N is the dataset size. MFCC results are two-dimensional data because they represent the time-frequency characteristics of a sound.

The two-dimensional MFCC results help train the CNN-LSTM model because the convolutional layer can capture spatial features. A CNN-LSTM hybrid model combines the strengths of both deep learning techniques in special cases, such as speech recognition, that utilizes MFCC feature extraction [30]. A 2D-convolutional layer produces a feature map by applying a convolutional filter to two-dimensional information such as a spectrogram. The formula is as follows:

$$y = f(W * x + b) \tag{5}$$

where $*$ is the convolution operation, W is the weights neuron, and b is the bias neuron. Neurons in LSTM can capture temporal dependencies in sequential data because they have memory in their cells (h_t). In this case, LSTM captures sequential dependencies in the feature map resulting from the 2D convolutional layer. Here are the formulas involved:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b) \tag{6}$$

where σ is the activation function, W_x is the input weights, W_h is the state weights, and b is the neuron bias.

Deep learning training requires several hyperparameter tuning to obtain a model that has optimum performance [31]. In this research, we carry out tuning to obtain values for the optimum model. Table 1 summarizes the hyperparameters we set. The important and influential hyperparameters are dropout rate, optimizer, learning rate, epochs, and batch size. The training and validation comparison curve can show whether there is overfitting or not. These hyperparameter settings are achieved through iterative empiric tests, which also involve the three other benchmark methods: LSTM, simple RNN, and GRU. Therefore, the three benchmark methods also use the same hyperparameter settings.

Table 1. CNN-LSTM Tuned Training Hyperparameters.

Variable	Value
Train:Test Data	80%:20%
Drop Out Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Epochs	150
Batch Size	300
Validation Split	10%

As a feature generator, we benchmark MFCC with LPC. LPC performs feature extraction on sound signals by representing the spectral envelope of the information [32]. LPC looks for correlations between each sample in a speech signal and a linear combination of previous speech signals. This method estimates the parameters of the linear model using techniques such as autocorrelation or covariance, which are then used to predict future samples. We compare the performance of the MFCC and LPC feature results with MDI, a technique for determining nodes in a random forest [33]. MDI can measure feature performance because MDI measures the average decrease in the Gini index of a feature during training [34]. The greater the value of an MDI, the greater the feature's contribution to a machine-learning model [35].

LSTM is a recurrent neural network method that captures temporal dependencies in sequential data. We benchmark LSTM with two other RNN methods namely simple RNN and GRU. Unlike LSTM which uses three gates (input, output, and forget gate), simple RNN uses only one number of gates [36]. The disadvantage is that simple RNNs have difficulty organizing information in sequential data, making it difficult to capture long-term dependencies. Compared to LSTM, which has three gates, GRU only has two gates: the reset gate and the update gate [37]. This makes GRU able to capture long-term dependencies like LSTM, on the other hand, simpler like RNN [38]. With a smaller number of gates, GRU and simple RNN have faster training times than LSTM [39].

3. Results and Discussion

3.1 *Results*

We downloaded the Synthetic Speech Commands Dataset from Kaggle uploaded by Johannes Buchner. The dataset is 1.77 GB in size and contains 83,700 WAV files. There are 30 utterances in the dataset, of which we selected 7: “left,” “no,” “off,” “on,” “other,” “right,” and “yes,” where “other” contains a mix of several utterances that we combined. We perform pre-processing on the dataset consisting of Numpy array conversion, channel separation, and normalization. Then we perform DSP consisting of a Fourier transform and a low-pass filter.

Figure 4 shows the Fourier transform of a speech dataset sample. The x-axis shows the frequency component of the signal, in units of Hertz (Hz). The frequency ranges between -0.5 Hz and 0.5 Hz, which indicates that the speech has negative and positive frequencies. The Fourier transform usually shows symmetric results between the negative and positive components. The Y-axis shows the amplitude of each frequency component whose units are decibels (dB). The highest peak amplitude reaches up to 100 dB, where there are several sub-peaks, which indicates that the speech has several dominant frequencies. The highest peak amplitude indicates the formant, which has significance in phoneme recognition. The center frequency is at 0 Hz, but this is a result of normalization. The original sound centered at 0 Hz indicates a low sound.

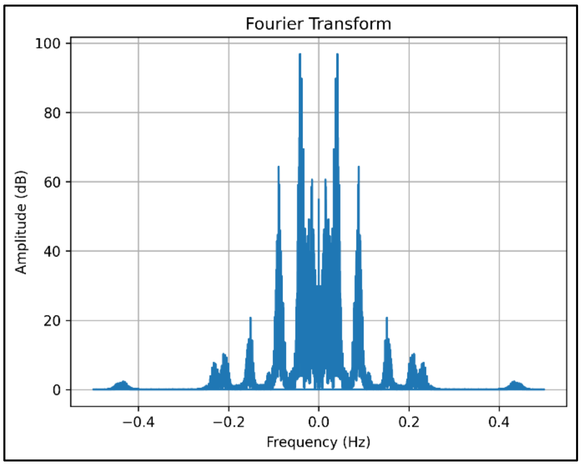


Figure 4. The Fourier Transform of a Speech Dataset Sample.

Figure 5 shows a comparison of the raw normalized speech signal sample (top image) with the filtered results of the speech signal (bottom image). In the original signal, it can be seen that there are signals with high amplitude. After filtering, the signal no longer prevails. This shows that a signal with a high amplitude has a high frequency. Because high-frequency signals are filtered by a low-pass filter, they are removed from the original signal. In accordance with its function, low-pass filtering produces a smoother sound signal. High-frequency signals that are unneeded in speech recognition are removed in this process.

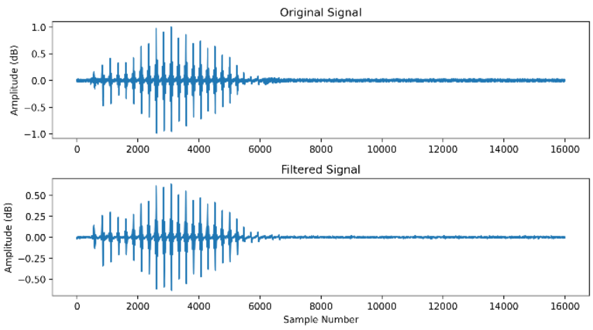


Figure 5. The comparison of a sample speech signal before and after low-pass filtering.

After performing DSP, the next step is to perform MFCC feature extraction. Figure 6 shows the time-varying spectral features of a speech sample in the dataset. The x-axis shows time, whereas the duration shows that the speech occurs in one second. The Y-axis shows the central coefficient, where a low coefficient shows broader spectral features and a high coefficient captures better detail. The plot shows that the lower coefficients have higher values, indicating that speech energy is concentrated in the lower spectral region, which is common in human speech. Changes in the central coefficient over time show that many salient features can be captured in a speech.

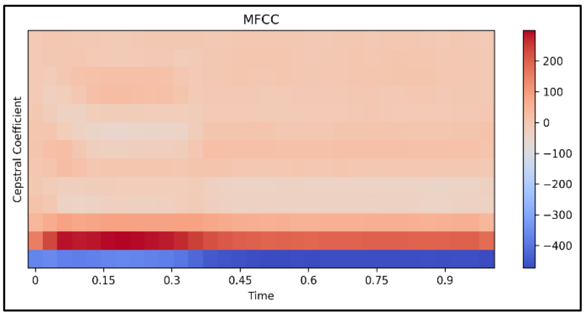


Figure 6. The MFCC of a sample speech data.

We benchmark the performance of the MFCC feature with LPC. Figure 7 shows the comparison of MDI scores between the two features. Each feature is represented by 13 features, each of which is extracted from the speech dataset. The bar plot is on a log scale, meaning the graph's y-axis increases exponentially. The average MDI score of MFCC features is 0.07686, while the average MDI score of LPC features is 0.00006. The feature performance of MFCC is notably higher than that of LPC. This shows the superiority of MFCC in capturing relevant information.

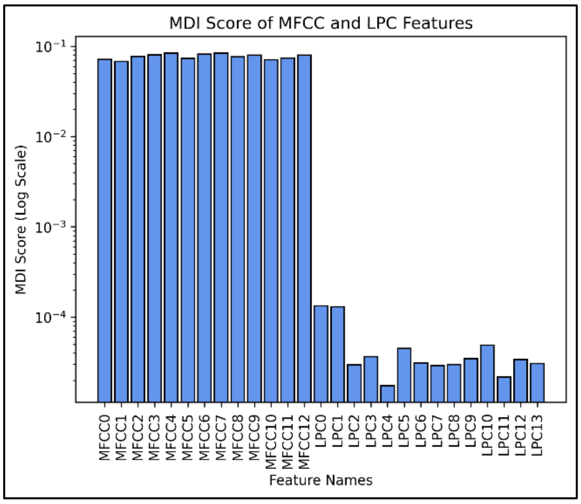
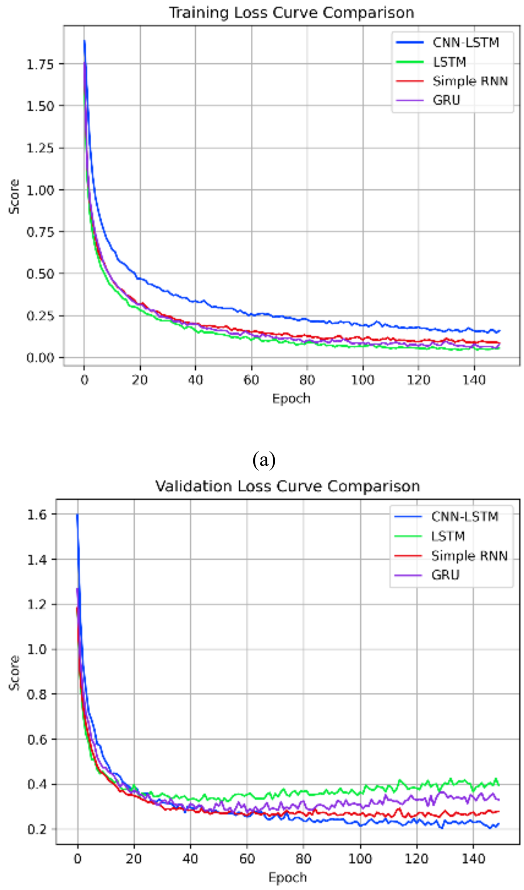


Figure 7. MFCC and LPC features comparison based on MDI score in log scale.

In the next testing step we compare CNN-LSTM with LSTM, simple RNN, and GRU. Comparing the training curve and validation curve can see whether there is overfitting or not. Figure 8 shows the curve. The four methods show good learning rates. In training, CNN-LSTM has the worst plateau. However, in validation, LSTM, simple RNN, and GRU show overfitting, while CNN-LSTM does not. Validation loss curve shows that CNN-LSTM has the best plateau among all methods.

Next, we compare CNN-LSTM with LSTM, simple RNN, and GRU in the next testing step. Comparing the training and validation curves enables the observation of possible overfitting occurrences. Figure 8 shows the curves. The four methods show good learning rates, visible through each loss's decreasing trend. In training, CNN-LSTM has the worst plateau. However, LSTM, simple RNN, and GRU show overfitting in validation, while CNN-LSTM does not. The validation loss curve shows that CNN-LSTM has the best plateau among all methods. These results should reflect on further performance measurement comparisons.



(a)

(b)

Figure 8. Loss curve comparison of CNN-LSTM, LSTM, Simple RNN, and GRU (a) Training Data (b) Validation Data.

The next test in benchmarking the performance of CNN-LSTM as speech recognition is to compare the method's accuracy, precision, recall, and f1-score with LSTM, simple RNN, and GRU. Figure 9 shows a bar plot comparing the performance of the four methods. CNN-LSTM has the best accuracy, precision, recall, and f1-score compared to the other three methods, with a value of 0.92 for all four metrics. Simple RNN is the second-best method, with a value of 0.91 for all four metrics. GRU is the third-best method with accuracy, precision, and an f1-score of 0.90, followed by a recall of 0.89. The method with the worst performance is LSTM, which has a performance of four metrics with a value of 0.89.

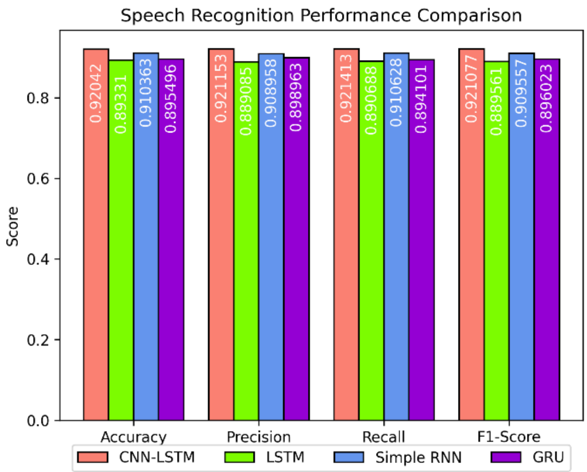


Figure 9. Bar plot showing performance comparison of speech recognition methods.

Speech recognition in this research classifies six types of commands for smart mirrors, namely "left," "right," "off," "on," "no," and "yes." In this test, we compared the performance of CNN-LSTM in classifying the six commands and non-commands ("other"). Figure 10 shows the confusion matrix. The label with the best precision is "no," with a value of 0.940. Conversely, the label with the worst precision is "other," with a value of 0.880, where 18 of the labels predicted to be "other" were actually "on."

As mentioned earlier, the Kaggle speech dataset contains 30 utterance labels, and we believe six of them are useful for the speech command of our smart mirror: “left,” “right,” “yes,” “no,” “on,” and “off.” We create a seventh label called “other” that contains a mix of several utterances that we combined. Based on the speech recognition test per label, “left” is the second label with a precision below 0.900; now, 26 of the labels predicted as “left” should be “other” or “yes.” Then the label with the best recall is "off," with a value of 0.963. In contrast, the label with the worst recall is “other,” with a value of 0.839; some 47 of the labels that should have predicted “other” instead predicted “on,” “off,” or “left.” Finally, the label with the best f1-score is “off,” with a value of 0.949. Meanwhile, the label with the worst f1-score is "other," with a value of 0.859.

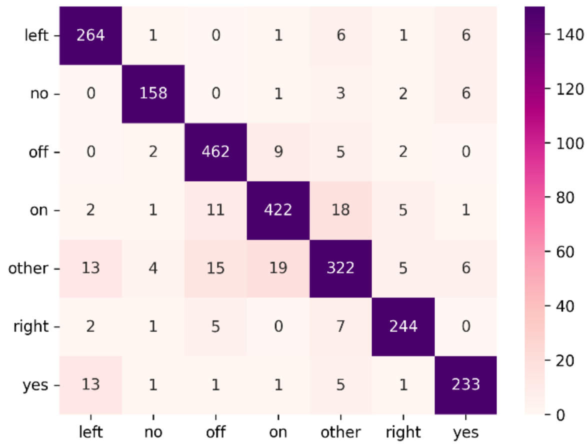


Figure 10. The confusion matrix showing the classification performance of CNN-LSTM on seven types of command speeches.

Finally, we designed a smart mirror that can display time and date, weather, and news updates, and use voice commands for control. We use Flask for API requests and Tkinter for GUI development. Figure 11, Figure 12, and Figure 13 show the display of three menus, news update, weather, and time and date, respectively. The speech commands "left" and "right" are useful for navigating the display between the three menus. Then the commands "yes" and "no" are practical for updating each menu. Finally, the "on" and "off" commands are instrumental in activating and deactivating the smart mirror.

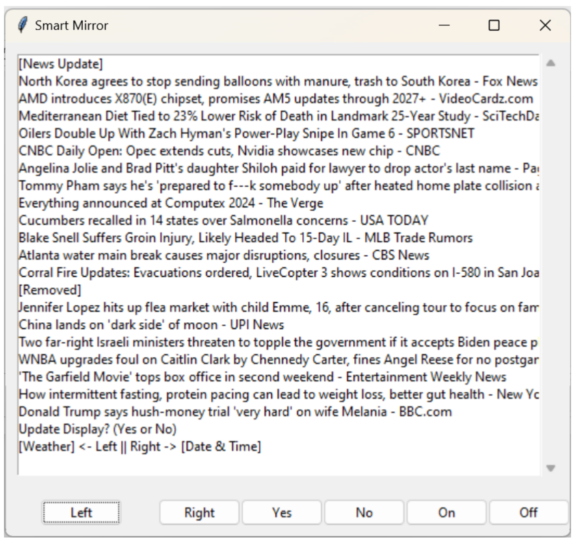


Figure 11. The smart mirror news update display.

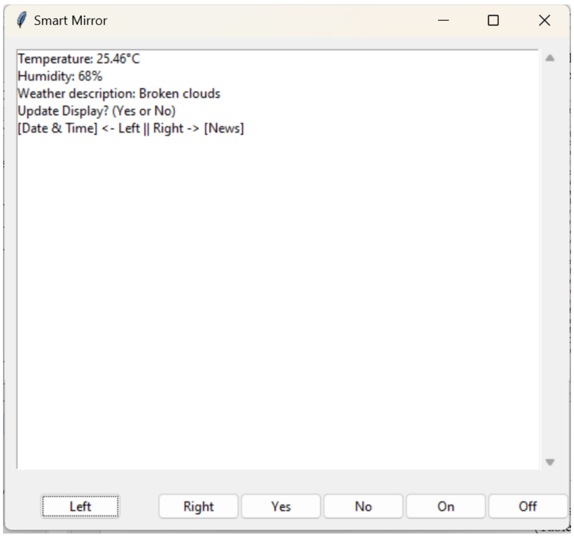


Figure 12. The smart mirror weather display.

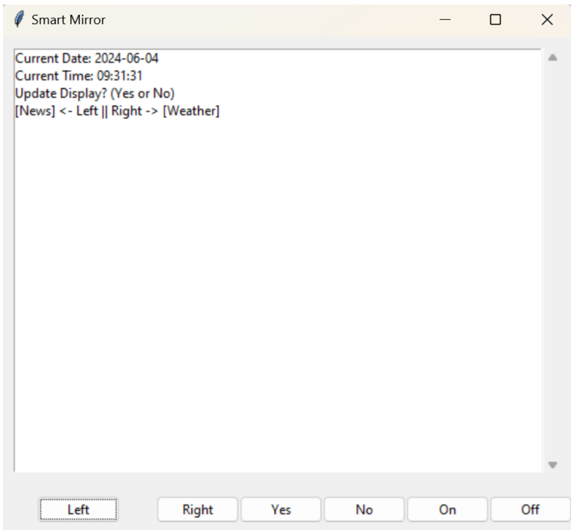


Figure 13. The smart mirror time and date display.

3.2 *Discussion*

Rauh *et al.* [40], in their paper, said that the speech frequency spectrum is sharp, high, and assembled in the middle. Meanwhile, the frequency spectrum of ordinary sounds appears even, spread out, without peaks. This is in line with our Fourier transform analysis of the speech dataset that we downloaded from Kaggle. The sounds we analyzed had the highest peak amplitude reaching up to 100 dB, where there were several sub-peaks, which shows that the speech had several dominant frequencies.

Ahmad *et al.* [41] mentioned that a low-pass filter removes high frequencies in sound thereby increasing the accuracy of speech recognition. High-frequency sounds in WAV speech files are usually background noise. This is in line with what we tested. In our study, before the low-pass filter, the amplitude range of normalized speech reached up to 1.0 dB, whereas after the low-pass filter, the range decreased to up to 0.5 dB, which shows that the removed high frequencies have high amplitude.

MDI is a method that has been widely used in the machine learning process in existing research. Altaf *et al.* [42] used MDI as feature selection in ensemble voting for disease diagnosis. Then Sandri *et al.* [43] used MDI to improve the performance of random forests and gradient boosting by analysing and correcting biases in these ensemble models. In this study, we use MDI to compare feature performance between MFCC and LPC, where MFCC has an average MDI score of 0.07686, while the average MDI score of LPC features is 0.00006. The results of this comparison was instrumental on defining high-quality features in speech recognition. Our research contribution is MDI as a feature comparison method between MFCC and LPC feature extraction.

Several studies have used CNN-LSTM for speech recognition with various case studies. Alsayadi *et al.* [12] used the hybrid model for Arabic speech recognition by looking at the effect of diacritics on speech recognition abilities. Kim *et al.* [44] used CNN-LSTM for speech recognition by looking at the influence of speech disorders in speech recognition. The research also found that the hybrid model performed better than LSTM for the case study. Our research contribution is a CNN-LSTM model for speech recognition with a case study of smart mirror speech command, where the hybrid model is the most optimal model compared to other methods.

Several studies have implemented smart mirrors with various functionalities. Paper [5] created a smart mirror with time plus date and weather functionality, then used voice commands. Papers [3] dan [4] built a smart mirror with features, namely time and date, weather, and news updates, and is equipped with voice commands. Paper [6] constructed a smart mirror with features, namely time, weather, and news updates, and is equipped with voice commands, where the voice commands used a cloud service, namely the Google Speech API. We implemented the speech command recognition feature using CNN-LSTM on a Raspberry Pi. This forms the concept of edge computing on smart mirrors and increases processing time on real-time-based smart things. Table 2 summarizes the comparison of features from related research regarding smart mirrors. Our research contribution is a smart mirror with the edge computing concept for speech command recognition.

Table 2. Features Comparison of Related Works on Smart Mirror.

Cite	Time & Date	Weather	News Update	Speech Command	Edge Computing
[3]	✓	✓	✓	✓	✗
[4]	✓	✓	✓	✓	✗
[5]	✓	✓	✗	✓	✗
[6]	✓	✓	✓	✓	✗
Proposed Method	✓	✓	✓	✓	✓

4. Conclusion

This research aims to apply CNN-LSTM to a smart mirror for speech command recognition to form the concept of edge computing. We benchmark CNN-LSTM with LSTM, simple RNN, and GRU methods. Our proposed smart mirror can detect six types of voice commands: "left," "right," "yes," "no," "on," and "off." The test results show that CNN-LSTM performs better than the three other methods with accuracy, precision, recall, and an f1-score of 0.92. The speech command with the best precision is "no," with a value of 0.940. Meanwhile, the command with the best recall is "off," with a value of 0.963. On the other hand, the speech command with the worst precision and recall is "other," with a value of 0.839. The contribution of this research is a smart mirror whose speech commands are carried out on the edge device with CNN-LSTM.

Acknowledgments

We thank Telkom University's Directorate of Research and Community Service (PPM) for continuing to support our research and funding the writing of this paper. We also thank Johannes Buchner for uploading the Synthetic Speech Commands Dataset to Kaggle.

References

[1] M. B. Satrio, A. G. Putrada, and M. Abdurohman, "Evaluation of Face Detection and Recognition Methods in Smart Mirror Implementation," in *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer, 2022, pp. 449–457.

[2] S. Bianco *et al.*, "A smart mirror for emotion monitoring in home environments," *Sensors*, vol. 21, no. 22, p. 7453, 2021.

[3] B. D. Majumder, R. Pratihar, R. Saha, and S. Ghosh, "Development of Interactive Smart Mirror for Implementation in College Environment," in *Proceedings of International Conference on Computational Intelligence and Computing*, J. K. Mandal and J. K. Roy, Eds., in Algorithms for Intelligent Systems. , Singapore: Springer Singapore, 2022, pp. 185–195.

[4] L. Tater, S. Pranjale, S. Lade, A. Nimbalkar, and P. Mahalle, "IoT based Assistive Smart Mirror with Human Emotion Recognition System," *Int J Eng Res TechnolIJERT*, vol. 9, pp. 381–385, 2020.

[5] M. M. Yusri *et al.*, "Smart mirror for smart life," in *2017 6th ICT International Student Project Conference (ICT-ISPC)*, IEEE, 2017, pp. 1–5. Accessed: May 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8075339/.

[6] M. Shakir *et al.*, "Smart Mirror Based Home Automation Using Voice Command and Mobile Application," *ICST Trans. Scalable Inf. Syst.*, p. 172102, Jul. 2018.

[7] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "EdgeSL: Edge-Computing Architecture on Smart Lighting Control With Distilled KNN for Optimum Processing Time," *IEEE Access*, vol. 11, pp. 64697–64712, 2023..

[8] J. Jo, J. Kung, and Y. Lee, "Approximate LSTM computing for energy-efficient speech recognition," *Electronics*, vol. 9, no. 12, p. 2004, 2020.

[9] J. Oruh, S. Viriri, and A. Adegun, "Long short-term memory recurrent neural network for automatic speech recognition," *IEEE Access*, vol. 10, pp. 30069–30079, 2022.

[10] X. Wang, F. Wang, B. He, and Y. Shan, "Wireless Standard Identification via Mel Frequency Cepstrum," *IEEE Commun. Lett.*, vol. 26, no. 11, pp. 2656–2660, 2022.

[11] Y.-H. Tu, J. Du, and C.-H. Lee, "2d-to-2d mask estimation for speech enhancement based on fully convolutional neural network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6664–6668. Accessed: May 30, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9054615/.

[12] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, and Z. T. Fayed, "Non-diacritized Arabic speech recognition based on CNN-LSTM and attention-based models," *J. Intell. Fuzzy Syst.*, vol. 41, no. 6,

pp. 6207–6219, 2021.

[13] N. Nalini, A. Shetty, and C. M. Abhishek, "The Mirror of the future: Building an InteractiveSmart Mirror with AI-based Virtual Assistant and Intruder Alert (Theft Detection)," 2023, Accessed: May 30, 2024. [Online]. Available: https://www.researchsquare.com/article/rs-2690711/latest

[14] V. B. Aanandhi, A. Das, M. G. Melchizedek, N. Priyadarsan, and A. Binu Jose, "Smart Mirror-Based Personal Healthcare System," in *Advances in Computing and Network Communications*, vol. 735, S. M. Thampi, E. Gelenbe, M. Atiquzzaman, V. Chaudhary, and K.-C. Li, Eds., in Lecture Notes in Electrical Engineering, vol. 735. Singapore: Springer Singapore, 2021, pp. 63–74.

[15] R. H. Aswathy, R. Saravanan, B. S. Sudhan, and D. Sasikiran, "Smart Mirror: A Magical Gadget for Diverse IOT Services," in *Ambient Communications and Computer Systems*, vol. 356, Y.-C. Hu, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds., in Lecture Notes in Networks and Systems, vol. 356. , Singapore: Springer Nature Singapore, 2022, pp. 435–446.

[16] V. M. Soppimath, M. G. Hudedmani, M. Chitale, M. Altaf, A. Doddamani, and D. Joshi, "The smart medical mirror-a review," *Int. J. Adv. Sci. Eng.*, vol. 6, no. 1, pp. 1244–1250, 2019.

[17] M. Wang, J. Marsden, and B. Thomas, "Smart mirror fashion technology for better customer brand engagement," *Int. J. Fash. Des. Technol. Educ.*, pp. 1–12, Aug. 2023.

[18] S. B. Handoyo, M. V. Setiawan, M. Valensius, and M. H. Widianto, "Future IoT Based on Smart Mirror: A Literature Review," *Int. J. Emerg. Technol.*, 2020, Accessed: May 30, 2024. [Online]. Available: https://www.academia.edu/download/89363806/44_20Future_20IoT_20based_20on_20Smart_20Mirror_20A_20Literature_20Review-3210-S._20Benson_20H.pdf

[19] I. C. A. García, E. R. L. Salmón, R. V. Riega, and A. B. Padilla, "Implementation and customization of a smart mirror through a facial recognition authentication and a personalized news recommendation algorithm," in *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, 2017, pp. 35–39. Accessed: May 30, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8334722/

[20] Z. Liang, Z. Liang, Y. Zheng, B. Liang, and L. Zheng, "Data Analysis and visualization platform design for batteries using Flask-based Python Web Service," *World Electr. Veh. J.*, vol. 12, no. 4, p. 187, 2021.

[21] V. Singh, Y. Rohith, B. Prakash, and U. Kumari, "ChatBot using Python Flask," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2023, pp. 1182–1185. Accessed: May 30, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10142484/

[22] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "Q8KNN: A Novel 8-Bit KNN Quantization Method for Edge Computing in Smart Lighting Systems with NodeMCU," in *Intelligent Systems and Applications*, vol. 824, K. Arai, Ed., in Lecture Notes in Networks and Systems, vol. 824. , Cham: Springer Nature Switzerland, 2024, pp. 598–615.

[23] F. J. B. Sanchez, M. R. Hossain, N. B. English, and S. T. Moore, "Bioacoustic classification of avian calls from raw sound waveforms with an open-source deep learning architecture," *Sci. Rep.*, vol. 11, 2021, Accessed: May 31, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8333097/

[24] D. Ye, "Digital Processing and Storage of Audio Data Based on Tag Information," in *2022 2nd International Conference on Networking, Communications and Information Technology (NetCIT)*, IEEE, 2022, pp. 383–387. Accessed: May 31, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10079082/

[25] A. G. Putrada, "Perancangan Software PDU Encoder dan PDU Decoder untuk Layer MAC WiMAX," *Indones. J. Comput. Indo-JC*, vol. 1, no. 1, pp. 24–36, 2016.

[26] M. A. Hasegawa-Johnson, J.-T. Huang, S. King, and X. Zhou, "Normalized recognition of speech and audio events," *J. Acoust. Soc. Am.*, vol. 130, no. 4_Supplement, pp. 2524–2524, 2011.

[27] H. Guan, H. Song, Y. Yang, and J. Zhang, "The fourier transform and its application in solving the equation," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 012056. Accessed: Jun. 02, 2024. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2012/1/012056/meta

[28] N. Bano, R. Pal, E. D. Verma, and S. Jain, "Literature Review of Low Pass Filters Based on CMOS for Biomedical Applications", Accessed: Jun. 02, 2024. [Online]. Available: https://www.academia.edu/download/110119264/ijraset.2023.pdf

[29] F. Hua and L. Li, "Sound anomaly detection of industrial products based on MFCC fusion short-time energy feature extraction," in *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, IEEE, 2022, pp. 861–864. Accessed: Jun. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10016076/

[30] S. F. Pane, J. Ramdan, A. G. Putrada, M. N. Fauzan, R. M. Awangga, and N. Alamsyah, "A Hybrid CNN-LSTM Model With Word-Emoji Embedding For Improving The Twitter Sentiment Analysis on Indonesia's PPKM Policy," in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia: IEEE, Dec. 2022, pp. 51–56.

[31] A. G. Putrada, N. Alamsyah, S. F. Pane, M. N. Fauzan, and D. Perdana, "Knowledge Distillation for a Lightweight Deep Learning-Based Indoor Positioning System on Edge Environments," in *2023 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, IEEE, 2023, pp. 370–375. Accessed: Jan. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10220996/?casa_token=f_dJRWL4VpkAAAAA:t-H9-HfnggTsdqhYXNcUObfQsD7d-2MnLTht-eaCMGNKrMt1r3eH5lrj5pkVmdgh9WUMTOGTkpk

[32] H. Shigeta, K. Komatsu, S. Oyabu, K. Matsuo, and S. Kurogi, "Analysis of Performance Improvement for Speaker Verification by Combining Feature Vectors of LPC Spectral Envelope, MFCC and pLPC Pole Distribution," in *Intelligent Systems Design and Applications*, vol. 418, A. Abraham, N. Gandhi, T. Hanne, T.-P. Hong, T. Nogueira Rios, and W. Ding, Eds., in Lecture Notes in Networks and Systems, vol. 418. , Cham: Springer International Publishing, 2022, pp. 220–230.

[33] E. S. Saputra, A. G. Putrada, and M. Abdurohman, "Selection of Vape Sensing Features in IoT-Based Gas Monitoring with Feature Importance Techniques," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, IEEE, 2019, pp. 1–5.

[34] A. G. Putrada, N. Alamsyah, S. F. Pane, and M. N. Fauzan, "Feature Importance on Text Analysis for a Novel Indonesian Movie Recommender System," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*, IEEE, 2023, pp. 34–39. Accessed: Jan. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10262504/?casa_token=NQvYjejOLKEAAAAA:xqEnEIgNxigp8UtAy7Fr8Mh67JiSNHe8aAiMY6iAnLdm-9Fh0UmxICw9LyZFGC2O5ljBqSt-CnY

[35] A. G. Putrada and N. G. Ramadhan, "MDIASE-Autoencoder: A Novel Anomaly Detection Method for Increasing The Performance of Credit Card Fraud Detection Models," in *2023 29th International Conference on Telecommunications (ICT)*, IEEE, 2023, pp. 1–6. Accessed: Feb. 09, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10374051/

[36] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "Recurrent Neural Network Architectures Comparison in Time-Series Binary Classification on IoT-Based Smart Lighting Control," in *2022 10th International Conference on Information and Communication Technology (ICoICT)*, IEEE, 2022, pp. 391–396.

[37] A. G. Putrada, N. Alamsyah, S. F. Pane, and M. N. Fauzan, "GRU-MF: A Novel Appliance Classification Method for Non-Intrusive Load Monitoring Data," in *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, Solo, Indonesia: IEEE, Nov. 2022, pp. 200–205.

[38] A. G. Putrada, N. Alamsyah, M. N. Fauzan, I. D. Oktaviani, and D. Perdana, "GRU for Overcoming Seasonality and Trend in PM 2.5 Air Pollution Forecasting," in *2023 Eighth International Conference on Informatics and Computing (ICIC)*, IEEE, 2023, pp. 1–6. Accessed: Jun. 03, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1038

1963/

[39] A. G. Putrada, "Gated recurrent unit for fall detection on motorcycle smart helmet with accelerometer sensor," *Indones. J. Comput. Indo-JC*, vol. 7, no. 3, pp. 21–32, 2022.

[40] A. Rauh, S. Tiede, and C. Klenke, "Comparison of different filter approaches for the online frequency analysis of speech signals," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, IEEE, 2017, pp. 605–610. Accessed: Jun. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8046897/

[41] R. Ahmad and S. Suyanto, "The impact of low-pass filter in speaker identification," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, 2019, pp. 133–136. Accessed: Jun. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9034592/

[42] I. Altaf, M. A. Butt, and M. Zaman, "Hard voting meta classifier for disease diagnosis using mean decrease in impurity for tree models," *Rev Comput Eng Res*, vol. 9, no. 2, pp. 71–82, 2022.

[43] M. Sandri and P. Zuccolotto, "Analysis and correction of bias in total decrease in node impurity measures for tree-based algorithms," *Stat. Comput.*, vol. 20, pp. 393–407, 2010.

[44] M. J. Kim, B. Cao, K. An, and J. Wang, "Dysarthric Speech Recognition Using Convolutional LSTM Neural Network.," in *Interspeech*, 2018, pp. 2948–2952. Accessed: Jun. 04, 2024. [Online]. Available: https://www.isca-archive.org/interspeech_2018/kim18e_interspeech.pdf