University of Essex
Department of Computer Science and Electronic Engineering

CE901-7-AU: MSc DISSERTATION

Comparative speech emotion recognition in multiple models and multiple datasets

Supasun Khumpraphan

Registration number:2110366

Supervisor: **Dr Cunjin Luo**

December 13, 2022

Colchester

Contents

1 Abstract **11**

2 Introduction **12**

3 Literature Review **14**

4 Background **17**

4.1 Data Description 17

4.1.1 Ravdess dataset 17

4.1.2 Crema-D dataset 17

4.1.3 Savee dataset 18

4.1.4 Tess dataset 18

4.2 Feature Extraction 18

4.2.1 Zero Crossing Rate 19

4.2.2 Root Mean Square 20

4.2.3 Mel Frequency Cepstral Coefficient 20

4.3 Exploratory Data Analysis 21

4.4 Machine learning 21

4.4.1 Logistic Regression 22

4.4.2 Support Vector Machine 23

4.4.3 K-Nearest Neighbors 25

4.4.4 Gaussian Naive Bayes 26

4.4.5 Stochastic Gradient Descent Classification 28

4.4.6 Decision Tree Classification 28

4.4.7 Random Forest Classification 30

4.4.8 eXtreme Gradient Boosting Classification 31

4.4.9 Multi-layer Perceptron Classification . . . 32
4.5 Deep learning . . . 35
4.5.1 Artificial Neural Network . . . 35
4.5.2 Deep neural network . . . 36
4.5.3 Convolutional neural network . . . 37
4.5.4 Recurrent Neural Network . . . 38
4.5.5 Long short-term memory . . . 39
4.5.6 Gated Recurrent Unit . . . 42
4.6 Train Test Split . . . 44
4.7 Feature scaling . . . 44
4.8 Cross-Validation . . . 45
4.9 evaluation . . . 45
4.10 Ensemble Learning . . . 46
4.10.1 Bagging . . . 46
4.10.2 Boosting . . . 47
4.10.3 Voting . . . 47
4.11 Early Stopping . . . 47
4.12 Learning Rate . . . 48
4.13 Hyperparameters . . . 48
4.13.1 Grid Search . . . 48
4.13.2 Random Search . . . 49

5 Methodology **50**
5.1 Data Preprocessing . . . 51
5.2 Exploratory Data Analysis . . . 51
5.3 Feature Engineering . . . 52
5.4 Train Test Split . . . 52
5.5 Model . . . 53
5.5.1 Machine Learning . . . 53
5.5.2 Deep Learning . . . 53
5.5.3 Ensemble Learning . . . 54
5.6 Metric . . . 54

6 Result **55**

6.1 predict only emotion labels . . . 55

6.1.1 Decision Tree . . . 55

6.1.2 Logistic Regression . . . 57

6.1.3 Random Forest . . . 58

6.1.4 K-Nearest Neighbors . . . 60

6.1.5 support vector machine . . . 61

6.1.6 Gaussian Naive Bayes . . . 61

6.1.7 eXtreme Gradient Boosting Classification . . . 62

6.1.8 Stochastic Gradient Descent . . . 63

6.1.9 Multi-layer Perceptron . . . 64

6.1.10 Convolutional neural network . . . 66

6.1.11 Long short-term memory . . . 68

6.1.12 Gated Recurrent Unit . . . 70

6.1.13 Deep neural network . . . 72

6.1.14 Artificial Neuron Network . . . 74

6.1.15 Convolutional neural network combine with Long short-term memory 76

6.1.16 Ensemble Learning with hard voting . . . 78

6.1.17 Ensemble Learning with soft voting . . . 78

6.2 predict gender and emotion labels . . . 79

6.2.1 Decision Tree . . . 79

6.2.2 Logistic Regression . . . 80

6.2.3 Random Forest . . . 81

6.2.4 K-Nearest Neighbors . . . 82

6.2.5 support vector machine . . . 83

6.2.6 Gaussian Naive Bayes . . . 84

6.2.7 eXtreme Gradient Boosting . . . 85

6.2.8 Stochastic Gradient Descent . . . 86

6.2.9 Multi-layer Perceptron . . . 87

6.2.10 Convolutional neural network . . . 88

6.2.11 Long short-term memory . . . 89

6.2.12 Gated Recurrent Unit . . . 91

6.2.13 Deep neural network 92
6.2.14 Artificial Neuron Network 94
6.2.15 Convolutional neural network combine with Long short-term memory 95
6.2.16 Ensemble Learning with hard voting 97
6.2.17 Ensemble Learning with soft voting 97
6.3 Compare model 98

7 Discussion **101**

8 Conclusion **104**

List of Figures

4.1 Structure of Zero Crossing Rate [38]. 20
4.2 The formula of a sigmoid function [5]. 23
4.3 The structure of a Support Vector Machine [33]. 24
4.4 relation with Entropy, Gini, Misclassification error [34]. 30
4.5 structure of Multi-layer Perceptron Classification[52]. 33
4.6 filter shifts by input and Calculation result which comes out as a Feature map[3] 37
4.7 confusion matrix [43] 46
4.8 early stopping technique [41]. 47

5.1 the research structure 50
5.2 Exploratory Data Analysis by histogram. 51
5.3 Exploratory Data Analysis by box plot. 51
5.4 wave of The Ravdess dataset, The Crema dataset, The Savee dataset, and The Tess dataset audio file 52

6.1 confusion matrix visualization Decision Tree model with default parameter . 55
6.2 confusion matrix visualization Decision Tree model with hyperpatameter . . 57
6.3 confusion matrix visualization logistic regression model with default parameter 57
6.4 confusion matrix visualization logistic regression model with hyperpatameter 58
6.5 confusion matrix visualization Random Forest model with default parameter 59
6.6 confusion matrix visualization Random Forest model with hyperpatameter . 59
6.7 confusion matrix visualization K-Nearest Neighbors model with default parameter 60
6.8 confusion matrix visualization K-Nearest Neighbors model with hyperpatameter 60
6.9 confusion matrix visualization support vector machine model 61
6.10 confusion matrix visualization Gaussian Naive Bayes model 62

6.11 confusion matrix visualization eXtreme Gradient Boosting model 62
6.12 confusion matrix visualization Stochastic Gradient Descent model with default parameter 63
6.13 confusion matrix visualization Stochastic Gradient Descent model with hyper-patameter 64
6.14 confusion matrix visualization Multi-layer Perceptron model with default parameter 64
6.15 confusion matrix visualization Multi-layer Perceptron model with hyper-patameter 65
6.16 Visualization of Loss Functions in Convolutional neural network model . . . 66
6.17 confusion matrix visualization Convolutional neural networks model 66
6.18 Visualization of Loss Functions in Long short-term memory model 68
6.19 confusion matrix visualization Long short-term memory model 68
6.20 Visualization of Loss Functions in Gated Recurrent Unit model 70
6.21 confusion matrix visualization Gated Recurrent Unit model 71
6.22 Visualization of Loss Functions in Deep neural network model 72
6.23 confusion matrix visualization Deep neural network model 73
6.24 Visualization of Loss Functions in Artificial Neuron Network model 74
6.25 confusion matrix visualization Artificial Neuron Network model 74
6.26 Visualization of Loss Functions in Convolutional neural network combine with Long short-term memory models 76
6.27 confusion matrix visualization Convolutional neural network combine with Long short-term memory model 76
6.28 confusion matrix visualization Ensemble Learning with hard voting model . 78
6.29 confusion matrix visualization Ensemble Learning with soft voting model . . 79
6.30 confusion matrix visualization Decision Tree model with default parameter . 79
6.31 confusion matrix visualization Decision Tre model with hyperpatameter . . . 80
6.32 confusion matrix visualization logistic regression model with default parameter 80
6.33 confusion matrix visualization logistic regression model with hyperpatameter 81
6.34 confusion matrix visualization random forest model with default parameter . 81
6.35 confusion matrix visualization random forest model with hyperpatameter . . 82

6.36 confusion matrix visualization K-Nearest Neighbors model with default parameter . . . 82
6.37 confusion matrix visualization K-Nearest Neighbors model with hyperpatameter 83
6.38 confusion matrix visualization support vector machine model with default parameter . . . 83
6.39 confusion matrix visualization support vector machine model with hyperpatameter . . . 84
6.40 confusion matrix visualization Gaussian Naive Bayes model with default parameter . . . 84
6.41 confusion matrix visualization eXtreme Gradient Boosting model with default parameter . . . 85
6.42 confusion matrix visualization stochastic gradient descent model with default parameter . . . 86
6.43 confusion matrix visualization stochastic gradient descent model with hyperpatameter . . . 86
6.44 confusion matrix visualization Multi-layer Perceptron model with default parameter . . . 87
6.45 confusion matrix visualization Multi-layer Perceptron model with hyperpatameter . . . 87
6.46 Visualization of Loss Functions in Convolutional neural network model . . . 88
6.47 confusion matrix visualization Convolutional neural network model . . . 89
6.48 Visualization of Loss Functions in Long short-term memory model . . . 89
6.49 confusion matrix visualization Long short-term memory model . . . 90
6.50 Visualization of Loss Functions in Gated Recurrent Unit model . . . 91
6.51 confusion matrix visualization Gated Recurrent Unit model . . . 92
6.52 Visualization of Loss Functions in Deep neural network model . . . 92
6.53 confusion matrix visualization Deep neural network model . . . 93
6.54 Visualization of Loss Functions in Artificial Neuron Network model . . . 94
6.55 confusion matrix visualization Artificial Neuron Network model . . . 95
6.56 Visualization of Loss Functions in Convolutional neural network combine with Long short-term memory model . . . 95

6.57 confusion matrix visualization Convolutional neural network combine with Long short-term memory model 96
6.58 confusion matrix visualization Ensemble Learning with hard voting model . 97
6.59 confusion matrix visualization Ensemble Learning with soft voting model . . 97

7.1 train model per minute in every model with just emotion labels 103
7.2 train model per minute in every model with gender and emotion labels . . . 103

List of Tables

6.1 comparative 9 models with only emotion label between default parameter and hyperparameter 98
6.2 comparative 9 models with gender and emotion between labels default parameter and hyperparameter 98
6.3 comparative 17 models with gender and emotion labels 99
6.4 comparative 17 models with only emotion labels 100

CHAPTER 1

Abstract

Understanding the feelings and emotions of the person we are talking to will allow us to have a better relationship with the person we are talk. Therefore, this research is to predict speech emotion recognition to understand human feelings better. There are two types of predictions: just the prediction of emotion and the prediction of emotion and gender. This research was performed by merging four datasets to solve the problem of imbalance and overfitting datasets. The four datasets were then combined and transformed from noise to numbers by Zero Crossing Rate, Root Mean Square, and Mel Frequency Cepstral Coefficient and predicted by 17 models to find the best model. The best model of machine learning from the experiment, Ensemble Learning with soft, had an accuracy of 89.82 percent for predicting only emotion and 88.64 percent for predicting emotion and gender. At the same time, the most accurate deep-learning model is the Convolutional neural network. The accuracy of predicting only emotion was 97.44 percent, and predicting emotion and gender was accurate at 97.35 percent. From this study, we can see that female voices were more accurate than male voices, fear voices were the least accurate, and surprise voices had the greatest predictive accuracy in model predictions.

CHAPTER 2

Introduction

Communication is an indispensable and essential aspect of human culture in today's society. Because we used to talk and exchange knowledge, making human beings have more progress. In addition, it increases the relationship, makes us understand each other, and participate in activities together. Understanding other people's thoughts and feelings makes us live with people happily and successfully with family, friends, and work. In order to adjust behavior according to the situation correctly and appropriately according to the situation to prevent quarrels and increase relationships with others. In some situations, more than verbal communication is needed to understand the other person entirely. However, seeing the speaker's gestures, faces, and mood will increase understanding and clarity. In business, it is vital to understand the sentiments of the employee of the company to work in order to get the best performance project. In addition, to talking persuasively to customers based on customer emotions. It will allow customers to want to use our company services. However, some services that do not see the customer's face make us unable to see their facial expressions and gestures, such as call centre systems which make understanding customer needs less effective and can lead to misunderstandings with customers. Therefore, companies should have sentiment and emotional analysis tools that understand the emotions of customers and analyze the diverse moods of customers in order to respond to customers effectively. This tool can also help assess the mood of service personnel. In studies where teachers teach, much effort is required to understand students to increase learning efficiency for students, especially in distance education. It is challenging to keep students interested in the subject they are

studying. Therefore, each teacher should formulate an appropriate lesson plan. The teaching should increase students' knowledge as much as possible but not be tedious or stressful until students do not want to study. In addition, every teacher should understand the feelings of every student. It will improve students' mental health and motivate them to study if the teacher does not have a proper teaching plan. It will give the students not enough motivation to study. The use of speech-emotion recognition can help solve educational problems by enabling teachers to understand students' feelings better and provide counseling. It helps students be more motivated to study because the teacher adjusts the lesson plan so that it is not too stressful. Help to improve the education system both near and far to be more effective [53]. In the medical field, the key components that make The patient have better physical health are that the patient is mentally healthy, cheerful, and encouraging, but in the current population that is growing in number and with a shortage of medical personnel to monitor emotions and counseling patients. Therefore, creating a robot to observe the behavior of patients is considered very important as it can replace medical personnel with a limit of doctors. A robot can observe the patient's emotional behavior throughout the day, analyze the patient's mood, and provide preliminary counseling. Difficult and complex cases can send messages to doctors to take over[46]. Regarding safety, car accidents result in 1.35 million deaths yearly due to many significant causes, such as drunkenness, recklessness, and anger. If we can check the mood of the driver, which is in an angry mood, it will send a warning that he should not drive the car and should wait for his mood to cool. In addition, there will be a robot to give advice to make feeling good and play music according to mood, which can reduce many traffic accidents. The research on speech emotion recognition is significant for the above reasons. Most of them use the model Convolutional Neural Network or Long Short-Term Memory or compare 2-3 models in deep learning, so it is still being determined which model is the best. This research will compare machine learning and deep learning models to find the best model.

CHAPTER 3

Literature Review

In research by Dipti D. Joshi and M.B. Zalte (2013), Recognition of Emotion from Marathi Speech using Mel-frequency cepstral coefficients and discrete wavelet transform algorithms. It studies emotion recognition performed in Marathi, one of the Indian languages. In this study, feature extraction from Pitch, formant, Mel-frequency cepstral coefficients, and Discrete Wavelet Transform, using a model support vector machine to predict angry, boredom, happy, sad, and neutral moods, with accuracy at 80.4 per cent[19].

In a study by Rajisha, Sunija and Riyas (2016), the spoken Malayalam language dataset was used. It was composed of 20 speech sounds, and each sentence had four tones: angry, happy, sad, and neutral, and was performed with feature extraction from Mel-frequency cepstral coefficients, Short Term Energy, and Pitch. Two models were used in Speech emotion recognition research studied the artificial neural network has 88.4 percent accuracy, and the support vector machine has 78.4 percent accuracy[39].

In a study by Albornozetal (2011), speech emotion recognition was used from the Berlin Emotional Database. The German language has ten sentences, ten speakers, and seven emotions neutral, anger, sadness, fear, boredom, happiness, and disgust. Feature extraction from the mean of log spectrum, Mel-frequency cepstral coefficients, and prosodic features was used in three models: the Hidden Markov model with an accuracy of 68.57 percent, the Gaussian Mixture Model with an accuracy of 63.49 percent, MLP with an accuracy of 66.83 percent[4].

A study by Bitouk et al.(2010) used speech from two datasets: the LDC Emotional Speech

Database, which is an English dataset. Using the voices of 7 people, four men and three women can collect up to 470 sentences. In addition, the emotional classification of 15 emotions includes boredom, shame, interest, elation, despair, pride, panic, neutral, sadness, happiness, contempt, fear, disgust, cold anger, and Hot anger. The Berlin Emotional Database use Feature extraction from Spectral features and modelling from a support vector machine. The experiment in LDC Emotional Speech Database has 46.1 percent accuracy, and Berlin Emotional Database has 81.3 percent[10].

Lee et al.'s research use voices from two datasets, including the AIBO DBUSB and IEMOCAP database. This research uses feature extraction from Mel-frequency cepstral coefficients, harmonics-to-noise, pitch, root mean square energy, Zero-Crossing Rate, and Decision tree as a model for this research. In this study, the AIBO dataset had an accuracy of 48.37 percent, and IEMOCAP had an accuracy of 58.46 percent[25].

In a study by Mao et al.(2014), a convolutional neural network was used to predict three datasets: SAVEE Database accuracy of 73.6 percent, Berlin Emotional Database accuracy of 85.2, and Mes database accuracy of 78.3 percent[28].

In a study by Rong et al. (2009), speech emotion recognition was predicted with a one-acted and one natural speech corpora in the china language dataset. Spectral features, Zero-Crossing Rate, and Pitch were used for feature extraction. A K-Nearest Neighbour model was used to predict the model of the 84 features. The average accuracy was 66.24 percent, the PCA and MDS accuracy was 61.18 percent, and the ISOMAP accuracy was 60.4 percent, while the best accuracy is done with the ERFTrees method. The method received an accuracy of 69.21 percent[42].

In the Siddique Latif study (2019), four emotional states were analyzed: angry, happy, sad, and neutral, using the IEMOCAP and MSP-IMPROV dataset and using the Convolutional Neural Network-Long Short-Term Memory-Deep Neural Network model to predict results. The prediction results with the Convolutional Neural Network-Long Short-Term Memory-Deep Neural Network model and the IEMOCAP dataset had an accuracy of 60.23 percent and Convolutional Neural Network-Long Short-Term Memory-Deep Neural Network With MSP-IMPROV dataset accuracy at 52.43 percent[23].

The research by Sagar k et al. (2016) was conducted with a model Naive Bayes classification. Mel-frequency cepstral coefficients, pitch, and energy were used as feature extraction using a self-generated dataset from males aged 18-30 years. The results were entirely accurate,

with an angry feeling at an accuracy of 81 percent, a happy feeling was accurate at 78 percent, and a neutral feeling was accurate at 77 percent, a sad feeling accurate at 76 percent[9].

The study of F. Noroozi et al. (2017) tested speech emotion recognition from the SAVEE dataset to predict seven feelings: anger, disgust, fear, neutral, happiness, sadness, and surprise. They used a random forest model to make their predictions. The accuracy for voice-based emotion was 78 percent[32].

Research by L. Kerkeni et al. (2018) used two datasets, the Berlin Emotional Database and the Spanish Database, in which both datasets were able to predict seven emotions: neutral, sad, surprise, fear, joy, disgust, anger, using the Recurrent Neural Network and MLR are a model for predicting feelings and uses Mel-frequency cepstral coefficients, and MS feature as features extraction which yielded satisfactory results in Spanish database using the Recurrent Neural Network model had an accuracy of 90.05 percent while the Berlin Emotional Database using MLR has an accuracy of 82.41 percent[21].

In the study of He, J., and Ren, L. (2021). used feature extraction 34 features: zero crossing rate one feature, energy one feature, energy entropy one feature, Mel-frequency cepstral coefficients 13 features, Chroma feature 12 features, chroma std one feature, spectral flux one feature, spectral roll-off one feature, spectral centroid spread one feature, spectral spread one feature, spectral entropy one feature and use XGBoost and Convolutional Neural Network BLSTM to predict. Three datasets were used to predict: EmoDB German emotional corpus accuracy 86.87 percent, CASIA Chinese emotional accuracy 74.17 percent, and EMA English emotional corpus accuracy 98.04 percent[15].

Jacob, A. (2017). examined speech emotion recognition using the Malayalam emotional speech database compiled from 10 speakers of 2800 speech files and tested on two models: Decision Tree has an accuracy of 93.63 percent, and logistic regression has an accuracy of 73 percent[18].

CHAPTER 4

Background

4.1 Data Description

This study uses four datasets, including Ravdess dataset, Crema-D dataset, Savee dataset, and Tess dataset, because if we use one dataset. It can be overfitting. So, this research use four datasets to prevent overfitting. These four datasets can combine 12162 records and seven emotions, including sad emotion 1923 records, happy 1923 records, angry 1923 records, fear 1923 records, disgust 1923 records, neutral 1895 records, and surprise 652 records.

4.1.1 Ravdess dataset

In this study, use the Ravdess dataset. It is a speech-only file that can be compiled from 12 female and 12 male speakers for a total of 24 speakers that collect 60 voices per speaker for a total of 1440. All speakers speak with a North American accent. This dataset can separate eight emotions: disgust, surprise, fear, anger, sadness, happiness, calm, and neutral. Except for neutral emotion, each can be separated into two levels: average emotional intensity and vigorous emotional intensity. In addition, we have two statements for speakers[17].

4.1.2 Crema-D dataset

This dataset is significant for predicting speech emotion recognition because this dataset has quite a lot of variety, which helps us to avoid overfitting. The crema-D dataset consists of

7442 voices from 91 speakers, 48 male speakers and 43 female speakers and ages of speakers between 20 and 74 from speakers on many continents and ethnicities, including Hispanic, Caucasian, Asian, American, African, and Unspecified. This dataset record speech of each actor with 12 sentences. Each sentence speaks a total of 6 emotions consisting Sad, Happy, Fear, Disgust, Anger, and Neutral. Each emotion has four intensity level includes Unspecified, Low, Medium, and High[12].

4.1.3 Savee dataset

SAVEE stands for Surrey Audio-Visual Expressed Emotion. The disadvantage of this dataset is that recordings do not have females. It has only male voices, resulting in an imbalance in this dataset. However, this dataset is characterized by very high-quality sound. This dataset should be used in conjunction with other datasets to prevent imbalance dataset issues. This dataset was compiled from four male postgraduate researchers students at the University of Surrey who are native English speakers aged 27-31, which can be distinguished from 7 emotions: neutral, surprise, sadness, happiness, fear, disgust, and anger. Each person will talk in about 120 sentences. This dataset has 480 voices[27].

4.1.4 Tess dataset

Tess stands for Toronto emotional speech set. It is one of the crucial datasets for speech emotion recognition because The speech emotion recognition dataset recorded primarily male voices. However, this dataset only recorded female voices as high-quality sound. Including this dataset solves the overfitting and imbalance problem caused by having too many male voices in another dataset. This dataset has two people to speak. Each person spoke seven voices: sadness, pleasant surprise, happiness, fear, disgust, anger, and neutral. Speakers must speak 200 sentences per emotion, which totals 2800 sentences[1].

4.2 Feature Extraction

In this research, speech signals are extracted from the feature extraction of different phonemes. This converts speech into numbers used in training data for the model. The learning process memorizes all the characteristics of each group of sounds. To compare when starting to

classify if the sound signal has the same characteristics. It will be able to identify which signal is in which group. Feature extraction also helps reduce the amount of data that needs to be analyzed and processed without data. All audio signals are compared. What will be analyzed will be only the characteristics that only extract and retain the essential properties of the data. This research will do three feature extractions: zero crossing rate, Root mean square and Mel Frequency Cepstral Coefficient[14].

4.2.1 Zero Crossing Rate

Zero Crossing Rate refers to the number of times the zero axis is cut across each frame. Zero Crossing Rate gives information about the number of zeroes of the signal prediction. That is when the signal changes rapidly. Then the number of zeros is more significant to receive the signal. That is, the signal must have high-frequency information. Similarly, a signal which is slow to change or it has less zero crossing rates. This would mean The signal must contain low-frequency information. Thus, the Zero Crossing Rate will method provides indirect information about the signal's available frequency signal. Where the Zero Crossing Rate is a fixed signal, it is defined as

$$z = \sum_{n=-\infty}^{\infty} |sgn(s(n)) - sgn(s(n-1))$$

where

sgn(s(n)) = 1 if s(n) $x \geq 0$

= -1 if s(n) $x < 0$

Correlation is adjusted for unstable signals. That is similar to the speaks voice, and the specified time is called Zero Crossing Rate is defined as

$$z(n) = \frac{1}{2N} \sum_{m=0}^{N-1} s(m) \cdot w(n-m)$$

When factor "2" is the divisor, there must be two values zero per signal cycle. In the case of speech, the nature of the signal changes with time beyond the estimated 2-3 milliseconds, which begins with speech. Not speaking and returning to vocalizations and other forms is beneficial information. The Zero Crossing Rate is required to be calculated. In the case of do not speak, the value of the Zero Crossing Rate was significantly high in the scope of the sound. Therefore, the Zero Crossing Rate can be used to differentiate the vocal and non-voice scope[6].

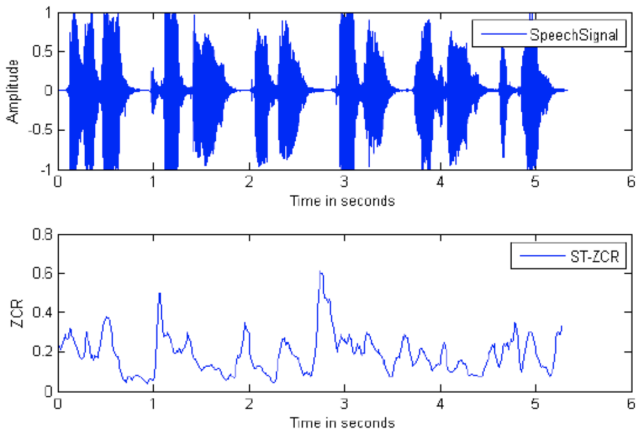

Figure 4.1: Structure of Zero Crossing Rate [38].

4.2.2 Root Mean Square

Root Mean Square(RMS) formed by the square of the function or the square root of the arithmetic mean of squares of values is a continuous-time waveform[14].

4.2.3 Mel Frequency Cepstral Coefficient

Mel Frequency Cepstral Coefficient (MFCCs) is the energy value of the spectral of a short-range audio signal. Mel Frequency Cepstral Coefficient is widely used in human speech recognition works. The inventors of this method were Davis and Mermelstein in 1980. The Mel Frequency Cepstral Coefficient works in harmony with the human auditory system, meaning that it only supports audio signals from 1 kHz and above. which is the range of frequencies humans can hear. This makes the Mel Frequency Cepstral Coefficient method ideal for speech recognition applications. The calculation of the Mel Frequency Cepstral coefficient is as follows. Firstly, Use the Windows function. Then, Calculate the spectral

energy value using the Fast Fourier transform (FFT) method. After that, Use a mel-filter bank. In addition, Use the discrete cosine transform (DCT). Finally, MFCCs are the heights of the resulting spectrum[30].

$$mfcc_i = \sum_{n=k-1}^{N} X_k \cos\left[I\left(k - \frac{1}{2}\right)\frac{\pi}{N}\right], i = 1, 2, ..., M$$

where

M = where M is the number of spectral coefficients.

k = 1,2, ..., N, is the log-energy effect of the K^{th} filter.

N = the number of triangular band pass fliters, the standard value is around 20.

4.3 Exploratory Data Analysis

Exploratory Data Analysis is understanding a data set, a step in the data analysis process that uses various techniques. To better understand the working data set, The purpose of audit data analysis is to. Firstly, a Better understanding of variables and their relationships between variables, such as in each Feature, as what kind of data is continuous or discrete, how vast is the range of data, how the data is distributed, and understanding. The relationship and how each Feature is linked, maximizing insights into our dataset and minimizing potential errors later in the process. Secondly, Explore and clean up the dataset, i.e. identifying outliers, missing values, how many missing data, isolating essential variables, and discarding variables that may distort the analysis.

4.4 Machine learning

Machine learning uses an algorithm that processes the data. Learn from data and lead to decisions depending on the input data. In machine learning using 11 models, including Decision Tree Classification, Logistic Regression, Random Forest Classification, K-Nearest Neighbors, Gaussian Naive Bayes, Support Vector Machine, eXtreme Gradient Boosting Classification, Stochastic Gradient Descent Classification, Multi-layer Perceptron Classifica-

tion, Ensemble learning with soft and hard voting to predict and compare multiple Speech Emotion Recognition model

4.4.1 Logistic Regression

Logistic Regression is one of the models that solve the classification algorithm used for discrete values used for predictive probabilities and the predictive analysis algorithm, which can classify the types of Logistics as regression three types. Firstly, Binary Logistic Regression will have only two possible answers, i.e. happy and sad. Secondly, Multinomial Logistic Regression will have more than two answers without ordering, such as happy, sad, and neutral. Finally, Ordinal Logistic Regression will have more than two answers with ordering, i.e. the 5-level happiness measure. Logistic Regression is similar to linear regression, but linear regression predicts a continuous value and uses a linear function. Logistic Regression predicts a discrete value and uses a cost function to define a sigmoid function, also known as a logistic function, to predict probability values. The sigmoid function has values from 0 to 1 with the following formula:

$$S(x) = \frac{1}{1 + e^{-x}}$$

where

S(x) = sigmoid function

e = Euler's number

The graph can be plotted in the picture below from the formula of a sigmoid function.

Where when it finds an outlier value in the sigmoid function. The logistic regression model will calculate the probability value. The result should be zero or one, which shifts the y-axis to the left or the right depending on the outlier value position. A Decision Boundary is also used to separate a class's probabilities. Which can separate the Decision Boundary has two types: Linear Decision Boundary and Non-Linear Decision Boundary[24].

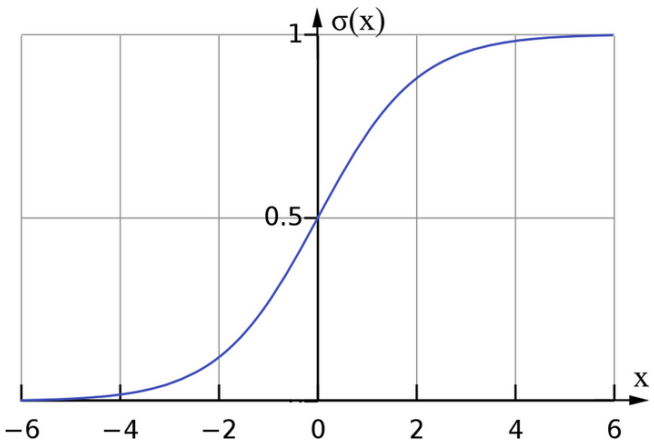

Figure 4.2: The formula of a sigmoid function [5].

4.4.2 Support Vector Machine

The Support Vector Machine principle is used to find the decision plane to divide data into two parts. It uses linear equations to divide two groups of fields from one another. Support Vector Machine is a type of machine learning that requires training data. It is a type of Supervised Learning that can do both Classification and Regression. The Support Vector Machine is based on Linear Classifier Computing, categorized for the best learning outcomes. (Discriminative Training) by learning from the statistics of the data. It works by finding the maximum Margin value of the Decision Hyperplane in split training data apart by using the function Map data from Input Space to Feature Space and creating a similarity measure function called Kernel Function on Feature Space. The objective is to minimize Error from prediction along with increasing Maximized Margin. Unlike conventional techniques, such as Artificial Neural Network, that only aims to minimize prediction errors, it is suitable for high dimensional data. Working principle of Support Vector Machine Data is written in the form of an ordered as follows:

$$\{(x_1, c_1), (x_2, c_2), (x_3, c_3), (x_n, c_n)\}$$

where

C_i = 1 or -1, it is defined as the segmentation of data X_i with a C_i of 1 and X_i with a C_i of -1

X_i = an actual vector dimension data value. When assigning this information to training data

This means that the segmentation of the data is accurate. Therefore, the segment line constructed as a general linear equation is $y = mx + b$, herein m is represented by W_t. to define an equation that can be expressed within it as follows:

$$W_t \cdot X + b$$

where

$W_t = theperpendicularvectoroftheslopemofthedividingline$

b = a constant derived from the y-axis values for each x data

The best group line is the group line. That gives the largest margin of the parallel lines extending from the separate line to the tangent to the closest data points of the two groups. We call that parallel line Parallel Hyper plane or can say It is the dividing line with the widest margin.

In Figure 4.3, at least one data point from both groups touching the most extended parallel line is called the support vector machine. this equation $W_t \cdot X + b = 1$ or $W_t \cdot X + b = -1$

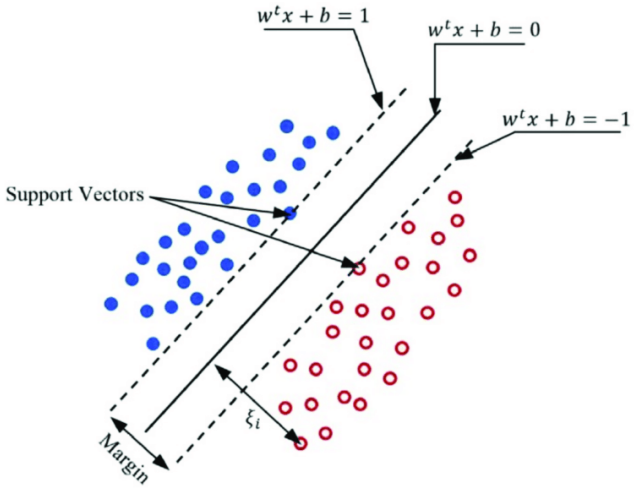

Figure 4.3: The structure of a Support Vector Machine [33].

Therefore, when the training data is segmented by any straight line, the widest distance of the two hyperplanes is extended until a point is found at the value of both groups is

$\frac{2}{|w|}$, where $|w|$ is the smallest. The data value for each X_i classified into any group can be considered from the following conditions:

If $W^t \cdot X_i + b \geq 1$ means that X_i is group 1 and if $W^t \cdot X_i + b \leqslant -1$, then X_i is group 2

Therefore, to extract any data points that were trained to filter that it was group 1 can be checked from this equation $c_i(W \cdot X_i - b) \geq 1$ when $1 \leqslant i \leqslant n$

Therefore, it can be written in short form. to find the smallest margin where the grouping can be calculated as this equation $Minimize_{w,b}|W|$ which can be checked from $c_i(W \cdot X_i - b) \geq 1$ when $1 \leqslant i \leqslant n$ [49]

4.4.3 K-Nearest Neighbors

K-Nearest Neighbors is a different class method for Classification. It uses the principle of comparing data of interest with other data to see how similar they are. If the data we are interested in is closest to the data. The system will give a similar answer to the nearest data. The behaviour does not use training data to create a model but instead uses all data. This technique will determine which class of data is similar to or close to by examining a certain number of data is K, and it is suitable for numeric data to determine how to measure the distance of each attribute in the data. The KNN model method has the following steps: firstly, We will need to set the value of k, which usually should be given as an odd number such as number 3, 5, 7, 9, etc. secondly, Bring the data to be classified to measure the similarities or differences of distance with all the data in the data set. The famous distance measure is Euclidean distance. The Euclidean distance has the following formula:

$$d(p,q) = \sqrt{\sum_{i=1}^{N}(q_i - p_i)^2}$$

where

p, q = two points in Euclidean n-space

q_i, p_i = starting for the initial point with Euclidean vectors

n = n-space

Thirdly, sort the data's spacing and consider a class of data by the number of k we define. Finally, The k group closest to our data becomes the class corresponding to the k group. The

advantage of KNN is that if decision conditions are complex, this approach can be used to create efficient models. However, the disadvantage of KNN is that it takes a long time to compute the model, and if the attribute is large, there will be errors in calculating the value. It can only calculate values for nominal data[35].

4.4.4 Gaussian Naive Bayes

In the world of machine learning, the key question is to find the mapping function (f) from our data (x) to the Target or Target class (Y) which from a probability perspective is P(Y | X) [54]. We have many classes The probability of occurrence in each class can make it not difficult and we can use with Bayes rule with this equation:

$$p(Y = y_i|X = x_k) = \frac{P(X = x_k|Y = y_i)P(Y = y_i)}{\sum_j P(X = x_k|Y = y_i)P(Y = y_i)}$$

where

X_k = the K^{th} values of all possible values of X

Y_i = the I^{th} values of all possible values of Y

The Bayes rule will find $P(Y|X)$ by estimating $P(Y|X)$ and $P(Y)$ from the training data first. Which uses the Bayes Rule is not good enough because it is not practical because of memory complexity. In case everything is only zero or one, it can considering from this equation:

$$P(X = x_i|Y = y_j)$$

where

i and j = the number of parameters we use up to $2(2^n - 1)$

n = number of features in X

It causes memory complexity with the exponential. Accordingly, we use the Naive Bayes classifier to solve memory complexity problems by mixing Conditional independence and

Bayes rule with this equation:

$$P(X|Y,Z) = P(X|Y)$$

where

X is independent of random variable Z when we know variable Y.

we can approximate $P(X|Y)$ more easily and efficiently if we have n features which means X consists of $X_1, X_2, ..., X_n$ then we get the following equation:

$$P(X|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

The parameter from $P(X|Y)$ that we need is only $2n$ (linear memory complexity), but everything can be survived with case 0 or 1. Therefore, the formula of $P(Y|X)$ also changes according to the conditional independence according to this equation:

$$p(Y = y_k|X_1, ..., X_n) = \frac{P(Y = y_k)\prod_i P(X_i|Y = y_k)}{\sum_j [P(Y = y_j)\prod_i P(X_i|Y = y_i)]}$$

we will choose the Y_k class that gives the highest probability of being answered every time we make predictions with new data based on this equation:

$$\hat{Y} \leftarrow \underset{Y_k}{\operatorname{argmax}}\, P(Y = y_k)\prod_i P(X_i|Y = Y_k)$$

We can see that the denominator is division missing because it is just a normalization term that makes the sum of all probabilities from all classes one, and it does not depend on Y_k. The Naive Bayes classifier is used to analyze the probability of something. That has never happened. By guessing from what has happened before, we will give two main approximations: $P(X|Y)$ and Prior $P(Y)$, where $P(X|Y)$ is for each class Y_k. The probability that features X_i will be one of all possible values. We will think of every feature, and each feature is independent of the others. We can define parameters as follows in this equation:

$$\theta_{ijk} = P(X_i = x_{ij}|Y = y_k)$$

We can estimate θ_{ijk} in 2 ways MLE and MLP. Estimated with MLE, which stands for Maximum likelihood estimation, using $len(D[condition])$ made from the number of rows from the data that made this condition true in this equation:

$$\hat{\theta}_{ijk} = \hat{P}(X = x_{ij}|Y = y_k) = \frac{len(D[X_i = X_{ij} and Y = y_k])}{len(D[Y = y_k])}$$

If approximated with a MAP abbreviated Maximum a posterior using a Dirichlet prior distribution which must have all the same parameters with this equation:

$$\hat{\theta}_{ijk} = \hat{P}(X = x_{ij}|Y = y_k) = \frac{len(D[X_i = X_{ij} and Y = y_k]) + l}{len(D[Y = y_k]) + lJ}$$

where

J = the total number of different values of the feature i.

l = the strength of the belief from the prior which is the proportion of the amount of data, so l = 1 is called laplace smoothing.

4.4.5 Stochastic Gradient Descent Classification

Stochastic Gradient Descent is a long-standing model of machine learning that just gained attention due to its broad learning context because Stochastic Gradient Descent solves the problem of predicting large and sparse data. This makes the Stochastic Gradient Descent a basic but effective predictive model used for Classification. However, Stochastic Gradient Descent requires a large number of hyperparameters, such as the number of iterations or regularization parameters and feature scaling, which significantly impacts the Stochastic Gradient Descent model[11].

4.4.6 Decision Tree Classification

A Decision Tree is a simulation of the way people make decisions. Each human decision will break the main problem into several sub-problems first in order to make the decision easier or will bring various factors related to decision-making or related to the main problem to create new questions or split into sub-problems and ask the problem again. and the reason why it is called Decision Tree because the idea is to branch or break the problem into smaller human problems before making a decision. It looks like the branching of a tree. The Decision Tree works in a rule-based model that generates an if-else condition from the data in a variable. In order to divide the data into new groups that best describe the Target, creating an If-else condition for each variable is defined with an Objective Function. The Model Decision Tree has several Objective Functions according to the Decision Tree type.

Decision Trees are divided into two types, Regression Trees and Classification Trees. In speech emotion recognition, we perform a Classification Tree. The Decision Tree used for making Classification uses Gini Impurity or Entropy as an Objective Function to find the best point in a split point. Gini Impurity is a measure of impurity or impurity in describing the target of a group that is divided from a variable. This means that the lower the impurity, the better the data can be split.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Calculating Gini Impurity

where

$\hat{p}$ = probabilities of the events of interest

$\sum_{k=1}^{K}$ = multiplying the sum of $\hat{p}$

Gini Impurity can be calculated by multiplying the sum of the events of interest (1 minus the probabilities of the events of interest).

After obtaining the Gini Impurity value of each group in every variable, the weighted Gini Impurity is evaluated to select the variable with the lowest Weighted Gini Impurity for decision-making first because it can best split the data.

$$GINI_{split} = \sum_{i=1}^{K} \frac{n_i}{n} GINI(i)$$

Calculating Weighted Gini Impurity

where

n_i = number of data of Class i

n = total number of data

Calculating Weighted Gini Impurity is taking the sum of the Gini values in the event of interest. Multiply by the number of data of Class I in the variable we are interested in and

divide by the total number of data in the variable we are interested in. Entropy is a measure of randomness, which measures uncertainty. In describing the target of a group separate from the feature, the lower the randomness value, the better the data is split.

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk}$$

Formular of Entropy

where

$\hat{p}$ = probabilities of the events of interest

$\sum_{k=1}^{K}$ = multiplying the sum of $\hat{p}$

The equation is similar to Gini Impurity, but Entropy is a multiplying log of the probability of the event we are interested in. Instead of multiplying by one minus the probability of the event we are interested in, Gini Impurity causes Entropy to range from 0 to 1 but Gini Impurity to range from 0 to 0.5[37].

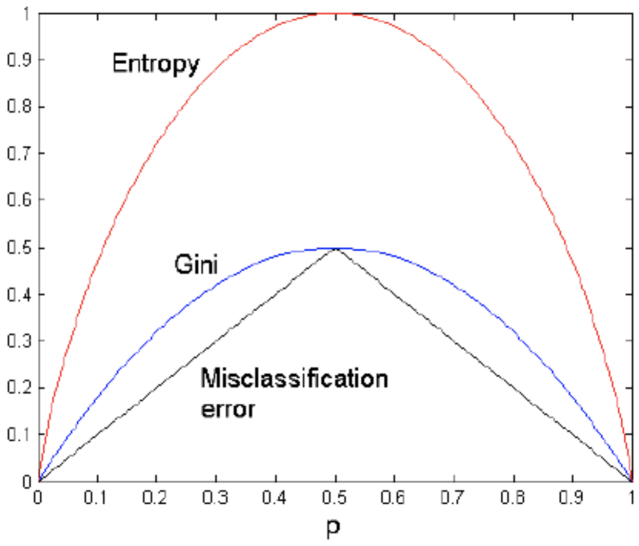

Figure 4.4: relation with Entropy, Gini, Misclassification error [34].

4.4.7 Random Forest Classification

Random Forest is formed by merging tree structures, in which the total forest discrepancy is converted to a limit value. It makes increases the number of trees in the forest. The total

discrepancy value is based on the strength of each tree, including the relationship between trees. It will use a random property-selection method for separating the nodes. It makes reduces the error value. This algorithm is very effective when implemented and used for sonic analysis and large projections. We can create a model that uses multiple decision trees to process. which has high accuracy. It can handle many data and is suitable for important data[48].

4.4.8 eXtreme Gradient Boosting Classification

eXtreme Gradient Boosting uses the Boosting method. This technique reduces variability and increases accuracy in the prediction of the classifier by using a method to reduce Bias and have a concept that allows a Weak Learner to work together until it can develop into a Strong Learner. A strong Learner has created a Weak Learner. Each can do this by increasing the weight of the false predictions each round and then doing new learning, which will make the model of the Classifier change with emphasis on more mistakes in the previous round. Once the number of weak learners was obtained enough, therefore, brought together. Build a Strong Learner in the Boosting step. It will think it is different from Bagging because Boosting is the implementation of weak classifiers with low accuracy for predicting the information we have. Then a new Weak Classifier fixes our error, where Aggregating sums are created as a new classifier. We will continue to do this (Recursive). Until the best model is obtained from the sum of the classifications if looking at the overall operation of Boosting is like working as a team. By bringing together the classifiers that are not very good until they can predict very complex data. However, the disadvantage of using Boosting is to run multiple times and sequences to get the desired model that is different from the same Bagging. That can then randomly data and train the model. However, the intelligence of eXtreme Gradient Boosting will be able to choose the type of Weak Learner or Weak Classifier, whether in tree form or linear form and many times, the Boosting technique Can predict very complex data more than using Bagging. eXtreme Gradient Boosting is based on the principle of Gradient Boosting, which is based on multiple classifier models. Models are there to help us find the answer. That is the Ensemble Classifier. which is a boosting group. However, it has improved to work faster and more efficiently. It can take full advantage of Multithread and adds regularisation variables to reduce the occurrence of a model that recognises data only one pattern or overfitting. Extreme Gradient Boosting uses the principle of creating

individual trees and being sequenced, where each input tree is an output from the previous tree. In principle, eXtreme Gradient Boosting creates individual trees. To reduce the error value caused by the previous tree using the Gradient Descend method and combining the results obtained. The advantage of eXtreme Gradient Boosting is that Bias and Variance are reduced because previous errors are fixed. Just the depth of the tree, just one layer, is enough to get the performance much better compared to Bagging Tree and Random Forest, which require more depth to provide similar performance[7].

4.4.9 Multi-layer Perceptron Classification

Multi-Layer Perceptron Neural Networks architecture is a Network for Prediction type often used for predictive tasks to help determine priorities. Multi-Layer Perceptron Neural Networks architecture or Multilayer Feedforward by Backpropagation learning was developed in the early years of 1970 by many sources and is working together to develop independently. Nowadays, Backpropagation architecture is also the most popular and efficient, and easy to use as a model for more complex multi-layer artificial neural networks. The technique of Backpropagation of this architecture has been used in many applications. It affects large network types in terms of shape and training methods because the main strength of the technique is the non-linear approach. That is suitable for solving unclear problems. Multi-Layer Perceptron Neural Networks are characterized by having one Input Layer and one Output Layer, and at least one Hidden Layer. The neural network feature has no theoretical limit on the number of hidden layers but, in practice, has only one or two. However, some complex solving operations require a minimum of four layers consisting of three hidden layers and one output layer, with each layer connected to a subsequent layer, as shown in Figure 4.5. Neurons are responsible for collecting input from other units, known as input. Log in where the input will be levelled according to the Weighted Connections value. Afterwards, before the sum value of each input unit is levelled. The weighted signal is output to the external via an axon, known as an output. Where the input sum value is activated to change the value again by the Activation Function or Transfer Function, perceptron can be represented by a mathematical model as follows:

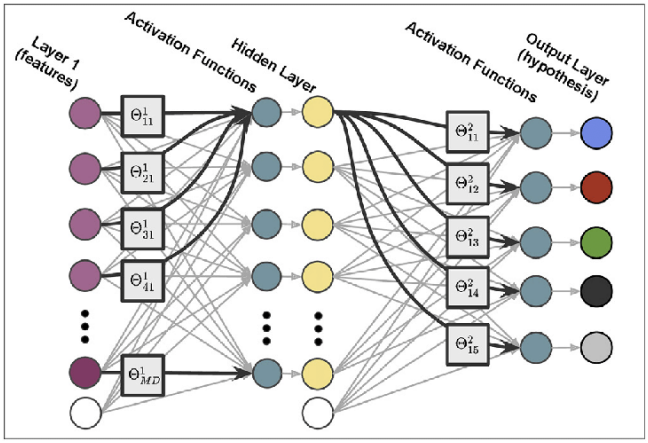

Figure 4.5: structure of Multi-layer Perceptron Classification[52].

$$o(X_1, X_2, ..., X_n) = \begin{cases} 1, & \text{if } w_o + w_1x_1 + w_2x_2 + ... + w_nx_n > 0 \\ -1, & \text{if } w_o + w_1x_1 + w_2x_2 + ... + w_nx_n < 0 \end{cases}$$

where

$x_1, x_2, ..., x_n$ = the inputs into the system

$w_1, w_2, ..., w_n$ = the weights of each input

where $x_1, x_2, ..., x_n$ are the inputs into the system and $w_1, w_2, ..., w_n$ are the weights of each input. output ($o(x_1, x_2, ..., x_n)$) of a function of the inputs as a linear sum weighted line. The weight determines which of the inputs (x_i) is important to the output configuration important will have the absolute value of much weight. On the contrary, less important ones are close to zero. If the sum equals zero, the output can be either one or minus one when given:

$$g(\overline{x}) = \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \overline{x}$$

Activation function in Perceptron

where

$\overline{x}$ = vector input

$\overline{w}$ = vector weight

The output of $\overline{x}$ equation will be as follows:

$$o(X_1, X_2, ..., X_n) = \begin{cases} 1, & \text{if } g(\overline{x}_i) > 0 \\ -1, & \text{if } g(\overline{x}_i) < 0 \end{cases}$$

The most popular Activation in Perceptron is the Bipolar Function, which shows the effect of 1 and -1, or a Binary Function, which shows the output of 1 and 0. A perceptron can have multiple inputs. If there are two inputs, the Perceptron is linear. In case of more than two inputs, Perceptron will be Hyperplane Decision Surface. Learning the Perceptron involves finding the appropriate weight vector (w) to classify the training data so that the Perceptron's output matches the training value based on the Perceptron Learning Rule.

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \leftarrow \alpha(t - o)x_i$$

formular of Perceptron Learning Rule

where

α = learning rate is a positive numerical constant

t = target output of the perceptron

o = output of the perceptron

Where α is the learning rate and is a positive numerical constant, t is the target output of the Perceptron, and o is the output of the Perceptron. Perceptron cannot learn some Activation. These activations are called Linearly Non-separable Functions, such as the XOR (Exclusive-Or) function, which is a limitation of Perceptron. Activation that can be separated is called a Linearly Separable Function, so Multilayer Perceptron or Multilayer Neural Network is designed to solve this problem. A multilayer neural network consists of several perceptrons. Let us connect in many ways. The connection style is Feedforward. and in the form of a layer of Perceptron. The layer that receives the data is called the Input Layer. The layer that performs internal processing is called the Hidden Layer. There may be several layers. The last layer is the layer that results in the network called the Output Layer. A multilayer neural network that is most used in research is a neural network that uses

algorithm Backpropagation, and a neural network is Multilayer Backpropagation Neural Network. Its added speciality is that it can create a nonlinear decision surface that is more isolated than a Linearly Decision Surface. The new learning rule is Delta Rule. It has the advantage that learning converges into a multidimensional plane with minimal error. It uses the principle of Gradient Descent. The delta law finds the weight vector that gives the least error value of the training sample by finding the mathematical Derivative. Therefore, an Activation function that can be found Derivative must be used, such as a Linear Function or a Sigmoid Function[47].

4.5 Deep learning

Deep learning is suitable for complex data, but it is more challenging to adapt to new data than Machine learning is suitable for data that can be defined features and clean data. Deep learning is a subset of machine learning, and machine learning is a subset of artificial intelligence. Deep learning nowadays is very capable. For example, cancer prognosis from X-ray images can be more accurate than doctors can differentiate the face of a person who can read traffic signs from photographs Separating speech from music. Play chess to win the world championship. to the creation of images of stars that do not exist. These abilities all have one point in common. That is, the input data to the deep learning algorithm is usually unstructured data. Also known as unstructured data, such as images, text, and audio, while conventional machine learning algorithms tend to be better suited to structured tabular data. Deep learning can learn from unstructured data better than machine learning[31].

4.5.1 Artificial Neural Network

Artificial Neural Networks, often abbreviated as Neural Networks or Neural Net, are one of the techniques of Data Mining, a mathematical model. For processing information with Connectionist computation to simulate the functioning of neural networks in the human brain to create a tool capable of learning Pattern Recognition and Knowledge Extraction, as well as the human brain's ability. The initial idea for this technique was to study the bioelectric network in the brain, which consists of neurons and synapses. Each neuron contains a nerve impulse end known as a dendrite. Which is the input, and the end of the nerve impulse is called the axon, which is like the output of a cell. These cells work with electrochemical

reactions. When stimulated by external stimuli or cells together, The nerve impulse travels through the dendrites into the nucleus, which decides whether to continue stimulating other cells. The nucleus then continues to stimulate other cells through its axon. Artificial Neural Networks are structured differently from the networks in the brain. However, still like the brain In the sense that Artificial Neural Networks are parallel clustering of subprocessors, and this connection is a crucial contributor to the intelligence of the network. Considering its size, the brain is significantly larger than Artificial Neural Networks. Including the neurons are more complex than the subunits of the network. However, these neural networks can still easily replicate essential brain functions such as learning. The working principle in computers Neurons consists of the same input and output, simulating that each input has a weight to determine the weight of the input. Each neuron has a threshold value that determines how large the total weight of the input must be. It can transmit output to other neurons. When these neurons are joined together, this function works logically like a chemical reaction in the brain. However, with computers, everything is just numbers. The function of Neural Networks is that when inputs come to a network, the inputs are multiplied by the weight of each leg. The resulting inputs on all the neuron legs are added together and compared to a predetermined threshold. If the value is greater than the threshold, the neuron will output. This output will be sent to the inputs of the other neuron connected to the network. The output will not be produced if the value is less than the threshold. We must know the weight and the threshold. For what we need for the computer to recognize is an uncertain value. Instead, a computer can be assigned to adjust those values by teaching it to recognize the pattern of what we want it to recognize, known as "backpropagation",. which is the reverse process of recognition. An algorithm is used for backpropagation. This is used to improve Network Weight. After each training data format is applied to the network, the network output is compared with the expected results. And then calculate the error value. This error value will be returned to the network for further correction of the score weight[22].

4.5.2 Deep neural network

DNN, or Deep Neural Network, is similar to an Artificial neural network. However, the difference is that a Deep Neural Network has more hidden layers to calculate complex data and solve problems with greater accuracy than Artificial Neural Networks[8].

4.5.3 Convolutional neural network

A Convolutional Neural Network is one of the bio-inspired neural networks in which the Convolutional Neural Network simulates the human vision of a sub-area and brings a group of sub-area to merge to see what exactly it is. Looking at a sub-area of humans, there are separate features of that sub-area, such as lines and colour contrasts. Humans know that this area is a straight line or a contrasting colour Because humans look at both the point of interest and the surrounding area. The convolutional neural network is a concept that is an excellent idea, but its complexity is a computational system that conforms to its concept and must support math. The calculations based on this concept are based on the same principle as Spatial Convolution in Image Processing. This calculation starts by assigning values in a filter or kernel that extracts the features used in object recognition. Usually, a filter or kernel can extract one feature of interest. So we need multiple kernels as well. to find a combination of multiple spatial features. Usually, the Filter by digital image is a two-dimensional grid with the size of the sub-area we want to consider.

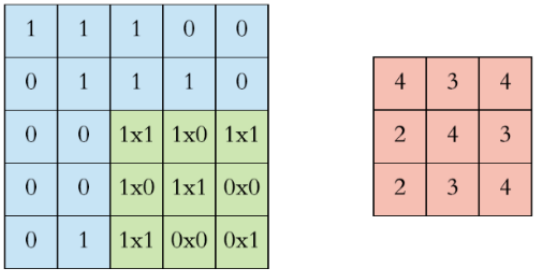

Figure 4.6: filter shifts by input and Calculation result which comes out as a Feature map[3]

From Figure 4.6, The centre position with the blue frame is the anchor placed on the input image's pixel. The filter is applied to the first pixel of the input image. Then it will be moved to shift on other pixels in the image, pixel by pixel, until all pixels in the image. We may not place filters on pixels near the frame because the filter will overflow out of the image When we move the filter until all the scrollable pixels are in the image. What we get is what is called a feature map. There is a Stride that determines how many steps we will move the filter. We can set Stride's value more if we want the computation of the feature to have less overlapping area. However, more Stride configurations will give us a smaller feature map. In addition, we can use padding as an area on all edges. These are the areas we usually fill in by adding 0 or other values to make a Convolutional Neural Network that Features a map. The result is

still the same size as the input because some edge issues can be crucial in affecting certain decisions. We, therefore, need to keep the edge feature as well, and there is a pooling that can shorten it. Two main types are the most popular: max pooling and mean pooling. Where Max Pooling is a filter that finds the maximum value in the area where the filter calculates the result, we will prepare a filter in the same way as a convolutional neural network is Feature Extraction, overlay the data and pick the highest value on that filter as a new result and will move the filter according to the specified Stride. The filter size of max pooling is commonly referred to as pool size, while mean pooling is similar to max pooling where max pooling filters the maximum value, while mean pooling is the filter that averages[3].

4.5.4 Recurrent Neural Network

Before we talk about Long Short-Term Memory and Gated Recurrent Units, we need to talk about the Recurrent Neural Network, the prototype of the Long Short-Term Memory and Gated Recurrent Unit. A recurrent neural network is a network that takes the output from the previous state as an input. The function is similar to the loop operation, which makes it look like a simple neural network with multiple and connects the output to a new network.

$$h(t) = x(t)W_{in} + h(t-1)W$$

Formular of RNN

where

$x(t)$ = output of neural network

$h(t-1)$ = output of previous neural network

W = Weight

The formula represents the use of the Output of $x(t)$ with the Output of $h(t-1)$, the previous Network Output, with a Weight of 2 modifiers of $x(t)$ and $h(t-1)$. The advantage of Recurrent Neural Networks is using the same data from previous networks, making them work well in time series data that take previous time data. Let us continue with the current time. The Time Series includes text data and audio data. However, in theory, the recurrent neural network is very good at solving long-term problems if the weight is chosen well, but the practice

could be more vital because of the weight selection. It also lost some important data. Long Short-Term Memory Network can solve this problem[29].

4.5.5 Long short-term memory

To solve the problem of RNN dealing with long data sequences, we can use Long Short-Term Memory if we look at RNN as a neural network that has simple memory inside to save the previous hidden state. It can be seen that LSTM has internal memory as well. However, what is better than RNN is that memory can also have a descriptor. When should write, forget for delete or allow read. It has a controller for read, write, or forget with analogue values. The work of The LSTM has variables that are Cell state stores the state of the memory cell in the LSTM. In addition, it has a Gate that controls data flow with analogue values that control when to read, write, or forget. Will see When information should flow in, out, or disappear altogether. The forget like clearing the old cell state to clear the area for receiving new information, but who decides whether to delete or delete is a function of a forget gate. The gate controls the flow of information inside. This is like a deciding factor. If forget gate value 0, then removes the original cell state. Nevertheless, if we forget gate value 1, we keep this cell state. In creating this forget gate, we will look at the incoming input data and combine it with the previous hidden state, which is similar to an RNN for decision making by using the sigmoid function as a decision maker. as the following equation:

$$f(t) = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Formular of Forget Gate

where

t = timestamp

f_t = Forget Gate at timestamp

x_t = input at timestamp

h_{t-1} = previous hidden state

W_f = weight matrix between Forget Gate and Input Gate

b_t = bias at timestamp

There is also a write operation where when new input data comes in. It is controlled by something called an input gate, which uses the sigmoid function to decide whether to allow updates or not, which first applies the input data value to the hidden state before that using this equation:

$$i(t) = \sigma(W_i \cdot [h_{t-1}, x_t] + b_t)$$

Formular of input Gate

where

t = timestep

i_t = Input Gate at timestep

W_t = weight matrix of sigmoid function between Input Gate and Output Gate

b_t = bias at timestep

In order to update, it uses called Input modulation gate to handle it. The equation is similar to the input gate, but uses the tanh function instead, which yields the value as a cell state candidate, giving us an idea of values we will update according to this equation:

$$\overline{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + B_c)$$

Formular of Input modulation gate

where

t = timestep

W_c = weight matrix of tanh function between cell state and network output

b_t = bias at weight matrix

After doing the input modulation gate, we update the cell state by integrating the forget gate, input gate and input modulation gate with the equation below:

$$C_t = f_t * C_{t-1} + i_t * \overline{C}_t$$

Formular of cell state

where

t = timestep

c_t = cell state infomation

f_t = Forgot Gate at timestep

i_t = Input Gate at timestep

C_{t-1} = previous cell state

$\overline{C}_t$ = value create by tanh function

After that, we will read, similar to the permission, whether to send the h_t value. Thus, we have an output gate to help decide. It will still use the same formula for the forget gate and the input gate, i.e. use the sigmoid function, the previous hidden state, and input data using the equations below:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

Formular of output gate

where

t = timestep

W_o = output gate at timestep

b_o = bias at weight matrix

h_{t-1} = output of previous LSTM

When getting the output gate. We will obtain the value of h_t for the following sequence using the equation below:

$$h_t = o_t * tanh(C_t)$$

Formular of LSTM output

where

t = timestep

h_t = output of LSTM

o_t = output gate at timestep

C_t = Cell state

It can be seen that if the output gate gives o_t a value of 0, then the value of h_t will be 0, and no value will be sent. At the same time, if o_t is one, then we calculate the value of h_t and Send it outside, such as allowing viewing the h_t value[16].

4.5.6 Gated Recurrent Unit

Gated Recurrent Unit is developed from LSTM, simplifying the operation of LSTM neural networks due to a large number of subunits in the cell, affecting analytical and predictive performance. The GRU simplifies the LSTM neural network's operation by reducing the cell subunits to only two parts: Update Gate and Reset Gate. The Update Gate is a subunit that uses the data calculated to determine the state of the cell for use in the next step by calculating every cycle that data comes in as below equation:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$

Formular of Update Gate

where

z_t = The value obtained from the update state.

σ = Sigmoid function

W_{xz} = weight value for calculating Input in Update Gate

x_t = input values to be calculated

W_{hz} = weight value for calculating Hidden State in Update Gate

h_{t-1} = Hidden state value obtained from the calculation in time unit before

b_z = bias value used in the calculation in Update Gate

Reset Gate is a unit used to determine how much of the state value obtained from the previous calculation should be stored as in the equation below:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$

Formular of Reset Gate

where

r_t = The value obtained from the update state.

σ = Sigmoid function

W_{xr} = weight value for calculating Input in Reset Gate

x_t = input values to be calculated

W_{hr} = weight value for calculating Hidden State in Reset Gate

h_{t-1} = Hidden state value obtained from the calculation in previous time unit

b_r = bias value used in the calculation in Reset Gate

In addition, we will calculate the value of Output and the Hidden State of GRU. Sub-units are calculated using the result value obtained from both Gates to calculate with the Tanh function. The value obtained from the Reset Gate determines whether to remember the original or remove the original value and control the amount of data with the value from Update Gate as below equation:

$$h_t = (1 - z_t) \cdot tanh(r_t \cdot W_{hh}h_{t-1} + W_{xh}x_t) + z_t h_{t-1}$$

Formular of Hidden State

where

h_t = The value obtained from the calculated hidden state

z_t = The value obtained from Update Gate

$tanh$ = Hyperbolic tangent function

r_t = The value obtained from the update state.

W_{hh} = weight value for calculating previous Hidden State

h_{t-1} = Hidden state value obtained from the calculation in previous time unit

W_{xh} = weight value for calculating Input

The operation of the GRU neural network is different from the LSTM neural network in that the cells of the GRU neural network are not stored. The status values obtained from the previous calculations were used in the GRU neural network analysis. which is shown in terms of efficiency. Compared to the LSTM neural network, the GRU neural network shows the advantage of reducing the number of training parameters. Make the model can work more quickly[13].

4.6 Train Test Split

We have to use train test splits because the model must work accurately with previously unseen data. It is called generalization. It is an essential concept of developing machine learning systems because if we have a system that works precisely only with the data it has seen before, It is like being a student who remembers the exam, enters the exam, and gets it right, only the questions that are the same but it can not deal with even slightly different problems when it comes to actual use When the actual data is found, the model will have unacceptably low accuracy performance. Causing overfitting problems. In the train test split step, we will divide data into two sets, with the train set to train the model and the test set to test for metrics after training. This is done to check the model's performance against data that has never been seen before[50].

4.7 Feature scaling

Feature scaling is scaling the scope of each cardinal feature's numeric data to the same range. Suitable for further processing Easy to enter formulas such as range [0, 1] or [-1, 1] results in a specified range called Data Normalization. Popularly done in steps, Preprocessing provides data Before feeding the model to train because the raw data we receive is diverse. Both the data type, data format, and scale. Data normalization will allow the Gradient Descent algorithm to converge faster. In this research, we will use Standardization or Z-Score Normalization that brings Feature data to adjust Mean = 0, and Standard Deviation = 1 in

python is StandardScaler[20].

$$z = \frac{X - \mu}{\sigma}$$

Formular of Standardization

where

X = observation value

μ = mean

σ = the standard deviation

4.8 Cross-Validation

Cross-validation is a popular method of research used to test the performance of the model due to the reliability of the results. Cross-validation will divide the data into several parts. This is usually represented by a value of k, also known as k-fold cross-validation, where each part contains a split of the same amount of data. After that, one piece of data is used to test the model's performance. Keep doing this until the allotted total amount is reached. We generally use k equal to 5 or 10 depending on the size of the dataset[40].

4.9 evaluation

Metrics are values calculated by comparing prediction and actual results to determine how accurate they are and how accurately our model predicts them. We will use the Confusion Matrix is an essential tool for evaluating the results of predictions or predictions predicted from the model we created. From Figure 4.7, it can be seen that True Positive is the prediction that matches what happened. In the case of the prediction, that is true and what happened is true. True Negative is the prediction that matches what happened. In the case of a prediction that is not true and what happens is not true. A false Positive is that the prediction does not match what happened. the prediction is true, but what happens is that it is not true. A false Negative is that the prediction is not true. The actual is a prediction that is not true. Nevertheless, what happens is true. The frequency represents the True Positive, True

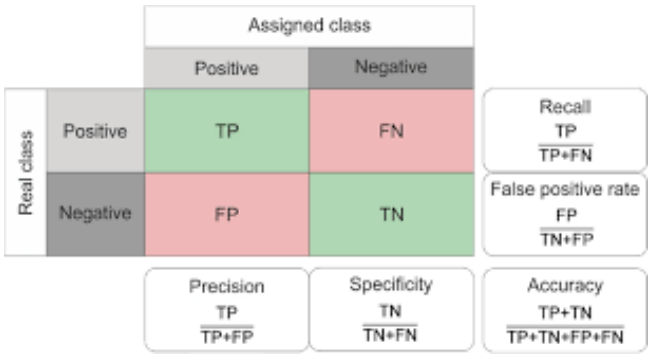

Figure 4.7: confusion matrix [43]

Negative, False Positive, and False Negative in the table. We can use the Confusion Matrix to calculate the effectiveness of our prediction model in many different values, including:

Accuracy calculated from $\frac{(TruePositive+TrueNegative)}{(TruePositive+TrueNegative+FalsePositive+FalseNegative)}$

Precision calculated from $\frac{TruePositive}{TruePositive+FalsePositive}$

Recall calculated from $\frac{TruePositive}{TruePositive+TrueNegative}$

The F1 score is calculated from $\frac{2*(Precision*Recall)}{Precision+Recall}$

Each metric has a different implementation. For example, Accuracy applies to balance data, but the F1 score applies to imbalance data. Also, writing in the python library sklearn can combine precision, recall, and f1 score with a single command with a classification report.

4.10 Ensemble Learning

Ensemble Learning is a viral technique used to develop models, significantly optimizing the model's performance. There are many popular techniques of Ensemble Learning[44].

4.10.1 Bagging

Bagging will create a tree of the same model algorithm to test against a subset of data that is divided from the whole data set. The prediction results of different models are then combined. Examples of this group of learning algorithms are Decision Trees, Random Forest, and Extra

Trees[51].

4.10.2 Boosting

Boosting generates multiple models, but the same model runs tests on the same set of data by doing an iteration test, looping, and adjusting the weight to improve the model's prediction results. This method is quite popular. Because it is flexible and can be used with all learning algorithms, it can reduce the bias error of the model well. Examples of these learning algorithms are AdaBoost and Stochastic Gradient Boosting[51].

4.10.3 Voting

Voting generates multiple models such as Decision Tree, SVM, and K-Nearest Neighbors and then runs them on the same data set to see the final prediction results of each model. Use a vote for identical, similar results as the final answer[51].

4.11 Early Stopping

Early stopping is the termination of training before the Optimiser converges to find the lowest loss, assuming that the more the training progresses, the higher the loss. The model becomes even more complex until it may be beyond being able to generalise well.

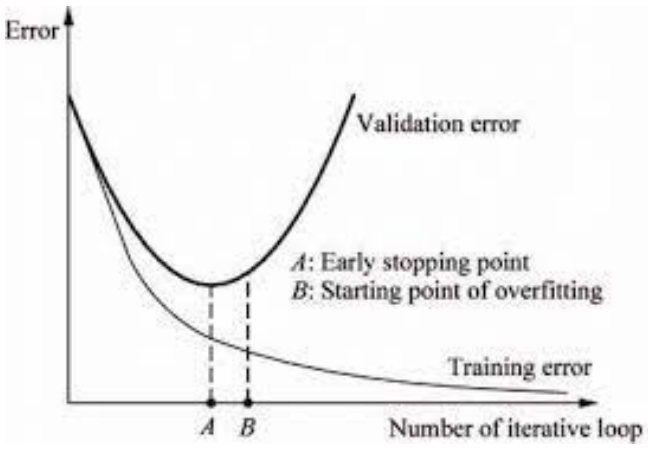

Figure 4.8: early stopping technique [41].

Figure 4.8 It can be seen that, normally, the Optimiser trains to get the lowest Train error, but sometimes the Test error starts to reverse to a higher value before reaching the lowest

point of the Cost function. Therefore, we should stop training. At the point where the Test error is the lowest, not the Train error is the lowest[36].

4.12 Learning Rate

Learning Rate is a hyperparameter that controls how much we should adjust the weight of the Neural Network in the training step. The higher the learning rate, the faster it is possible to train the target model. However, if the learning rate is too high, the model's training will skip the step until sometimes it may cause us to fall from the desired goal. If the learning rate is too low, the model's training will be very slow until it may not reaches the desired goal in the specified time. Therefore, tuning the learning rate is an essential step in model training. The optimal learning rate depends on the loss function, the model, and the training data. It is possible to have a single value that can be used appropriately in all cases. Nevertheless, the learning rate is typically around 0.1 to 0.01[26].

4.13 Hyperparameters

Hyperparameters are parameters that the user can define before the model learns. We need to adjust the parameters even though each model gives a default parameter that already exists because the default parameter is suitable for all datasets but not the best for all datasets. Adjusting hyperparameters will increase the model's accuracy or reduce the loss to the lowest by searching for hyperparameters currently popular in 2 types.

4.13.1 Grid Search

It is a technique used to find values. Hyperparameter that is easy to understand and straightforward, trying every set of predefined parameters. Moreover, evaluate the performance or accuracy of each model. It will try to create a model from the value of Every set of hyperparameters that works like a grid. All values are in the form of a matrix. Each set of parameters is taken into account and observed for correctness. All have been assessed. Models with the most accurate set of parameters are considered the best.

4.13.2 Random Search

Random Search works similarly to Grid Search, but instead of trying predefined parameters on every grid, RandomSearch randomly selects parameter values from the generated grid, so Random Search does not guarantee a model. The most efficient method is the same as Grid Search, but the method is highly efficient in real-world applications due to the minimal modelling time required.

CHAPTER 5

Methodology

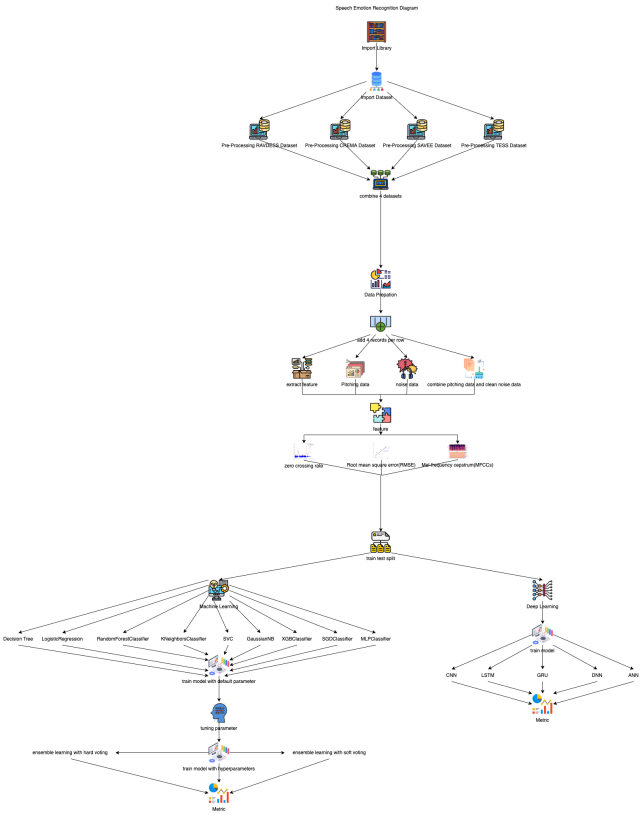

Figure 5.1: the research structure

5.1 Data Preprocessing

In this exploration, clean data and combining all four datasets Ravdess Dataset, Crema Dataset, Savee Dataset, and Tess Dataset, can get a total of 12162 records, which is enough to model both Machine Learning and Deep Learning. All 4datasets will be extracted using three columns: the location of the Audio file, emotion, and gender of the speaker. After that, we will visualize to see any problem with the dataset.

5.2 Exploratory Data Analysis

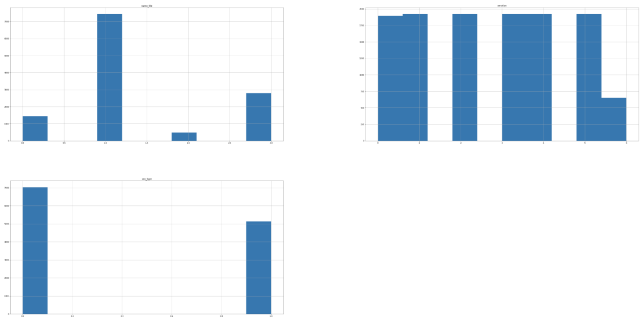

Figure 5.2: Exploratory Data Analysis by histogram.

This research uses Exploratory Data Analysis to explain the dataset. Although, dataset Crema has a lot more than dataset Ravdess, Savee, and Tess. However, It can be seen that in Figure 5.2. We combine four datasets to fix the imbalance problem because It makes this dataset more balanced. Although the feeling of surprise is much less than the other emotions but the other emotions are very balanced. In addition, female audio files have approximately 58 percent, and male audio files have approximately 42 percent, which is not much difference, which makes this dataset can fix imbalances and overfitting. After that, we use a box plot in Figure 5.3 to find outliers in every column. We note that this dataset does not have outliers in all columns.

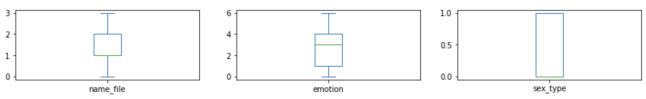

Figure 5.3: Exploratory Data Analysis by box plot.

5.3 Feature Engineering

In this topic, we will use two ways. First, we will remove the column's gender of the speaker and use only emotion to predict. In a second way, this research will use both emotion and gender to predict models. After that, we will use 3 Features: Zero Crossing Rate with 108 columns, Root Mean Square with 108 columns, and Mel Frequency Cepstral Coefficient with 2161 columns. Which combines three features, there will be a total of 2376 columns, and in this experiment, we can see from Figure 5.4. Most of the audio files start talking at 0.6 seconds or higher, with the first 0.6 seconds having no speech or emotion at all, and most of the audio data is less than 3 seconds. Thus we took the audio data in 0.6 seconds to 3 seconds to get the most talker voice and the model to be the most accurate.

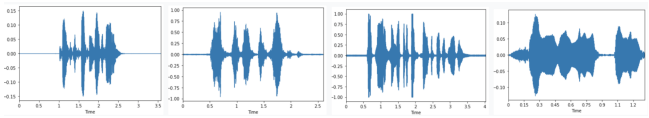

Figure 5.4: wave of The Ravdess dataset, The Crema dataset, The Savee dataset, and The Tess dataset audio file

After that, we will create four times more labels from the original to check the data according to 4 scenarios, namely the voice data from the talker without any adjustment, the voice data from the talker with noise, the audio data from the talker with pitching data but it is noise, the voice data from the talker comes with pitching data, and there is no noise. It increases the number of records from 12162 to records are 48648 records. After converting from audio file to number, we get 48648 rows and 2377 columns in case there is no gender label and 2378 columns in the case has a gender label. However, some audio data has less than 3 seconds of speech. This causes some data less than 3 seconds to be null values, so we replace the null value with 0 to clean the data.

5.4 Train Test Split

Then we will use the train test split by bringing the feature and label to use and set the random state to make the model predicted stable accuracy every round at the run model and divide the train set to 80 percent in the model prediction and the test set to 20 percent for the

test model with unseen data. In predicting the model, we must use feature Scaling by the method we use Standardization or in python is StandardScaler, to keep data in a given scope for increased accuracy. We will predict two ways those that are both gender and emotion label and those that are only emotional labels.

5.5 Model

5.5.1 Machine Learning

After we split the data, we will train a total of 9 machine learning models in the next step. For each model, we will train two times. First, we train each model using the default parameter. After that, we find the best parameter by Hyperparameters the parameter. This research parameter uses RandomizedSearchCV and gives ten parameters that are randomly selected. The best selected and given cv equal to 5 is 5-fold cross-validation that divides the data into five parts and is predicted and Finds the mean accuracy of 5 cross-validations. Although GridSearchCV finds every parameter, it will find more accurate parameters than RandomizedSearchCV. However, this research uses RandomizedSearchCV because the data we use for research is enormous, so it takes a long time to train and cannot be completed. Using RandomizedSearchCV is, therefore, the best choice for this research. After we have done the Hyperparameters, we will use the parameters to train each model, evaluating the model with Hyperparameters in each step, both the default parameter and the evaluation model with Hyperparameters of each model. We will do K-Fold Cross Validation by dividing the data into equal parts to train and validate for calculating average accuracy.

5.5.2 Deep Learning

In deep learning, we will train six models using call back to set Early Stopping so that the value is not overfitting and set the weight with the Learning Rate so that the value is not too high to cross the steps in the training model and it is not too low for a not too late time. We also determine the epoch equal to 50 and batch size equal to 32. The training data is used to train the model, and the test data is used to measure the model's validity. After that, we display loss and accuracy metrics with evaluate function to see the model performance. Then we make two comparisons: comparing the accuracy metric and validation accuracy in every

epoch and comparing the loss metric and validation accuracy in every epoch.

5.5.3 Ensemble Learning

After we train the model for both machine learning and deep learning, we take the best model but change the parameters for voting to make the model more accurate. If the model is a machine learning model, the workflow is the same as the machine learning model, but if the model is a deep learning model, the workflow is the same as the deep learning model.

5.6 Metric

In this step, we will predict the data and metrics model using a classification report, accuracy score, and confusion matrix to measure the accuracy of every model.

CHAPTER 6

Result

In making predictions, we will make two predictions, which are only emotion prediction labels and predict both emotion labels and gender labels.

6.1 predict only emotion labels

6.1.1 Decision Tree

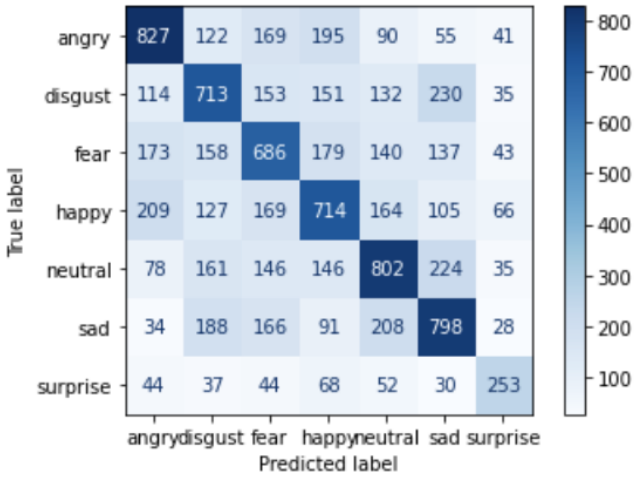

Figure 6.1: confusion matrix visualization Decision Tree model with default parameter

Figure 6.1 shows the confusion matrix in the Decision Tree model with default parameters that can be predicted angry emotion correct 827 labels accounted for 55.17 percent,

disgust emotion correct 713 labels accounted for 46.66 percent, fear emotion correct 686 labels accounted for 45.25 percent, happy emotion correct 714 labels accounted for 45.94 percent, neutral emotion correct 802 labels accounted for 50.37 percent, sad emotion correct 798 labels accounted for 52.74 percent, surprise emotion correct 253 labels accounted for 47.91 percent, The correct sum of 4793 labels gives an accuracy of 49.26 percent.

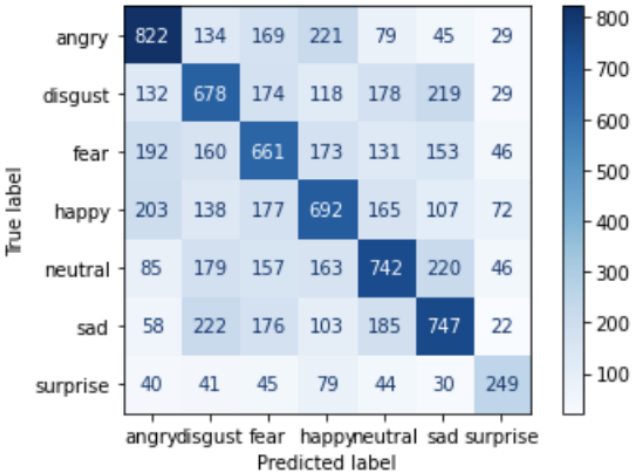

Figure 6.2: confusion matrix visualization Decision Tree model with hyperpatameter

While the figure 6.2 represents a Decision Tree model in hyperparameters that can predict angry emotion correct 822 labels accounted for 54.83 percent, disgust emotion correct 678 labels accounted for 44.37 percent, fear emotion correct 661 labels accounted for 43.60 percent, happy emotion correct 692 labels accounted for 44.53 percent, neutral emotion correct 742 labels accounted for 46.60 percent, sad emotion correct 747 labels accounted for 49.37 percent, surprise emotion correct 249 labels accounted for 47.15 percent, The correct total number is 4591 labels, giving an accuracy of 47.18 percent.

6.1.2 Logistic Regression

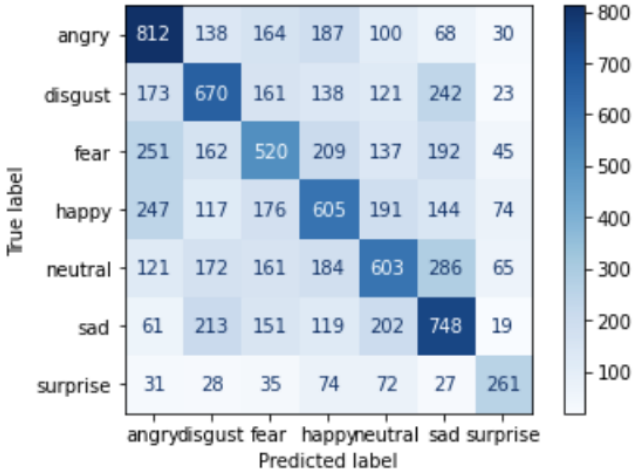

Figure 6.3: confusion matrix visualization logistic regression model with default parameter

Figure 6.3 shows the confusion matrix in the logistic regression model with default

parameters that can be predicted angry emotion correct 812 labels accounted for 54.16 percent, disgust emotion correct 670 labels accounted for 43.84 percent, fear emotion correct 520 labels accounted for 34.30 percent, happy emotion correct 605 labels accounted for 38.93 percent, neutral emotion correct 603 labels accounted for 37.87 percent, sad emotion correct 748 labels accounted for 49.43 percent, surprise emotion correct 261 labels accounted for 49.43 percent, The correct sum of 4219 labels gives an accuracy of 43.36 percent.

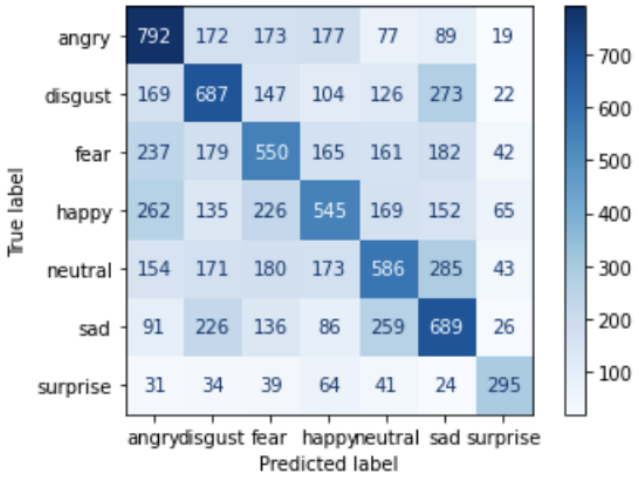

Figure 6.4: confusion matrix visualization logistic regression model with hyperpatameter

While the figure 6.4 represents a logistic regression model in hyperparameters that can predict angry emotion correct 792 labels accounted for 52.83 percent, disgust emotion correct 687 labels accounted for 44.96 percent, fear emotion correct 550 labels accounted for 36.27 percent, happy emotion correct 545 labels accounted for 35.07 percent, neutral emotion correct 586 labels accounted for 36.80 percent, sad emotion correct 689 labels accounted for 45.53 percent, surprise emotion correct 295 labels accounted for 55.87 percent, The correct total number is 4144 labels, giving an accuracy of 42.58 percent.

6.1.3 Random Forest

Figure 6.5 shows the confusion matrix in the Random Forest model with default parameters that can be predicted angry emotion correct 1307 labels accounted for 87.19 percent, disgust emotion correct 1170 labels accounted for 76.57 percent, fear emotion correct 983 labels accounted for 64.84 percent, happy emotion correct 1162 labels accounted for 74.77 percent, neutral emotion correct 1260 labels accounted for 79.14 percent, sad emotion correct 1218

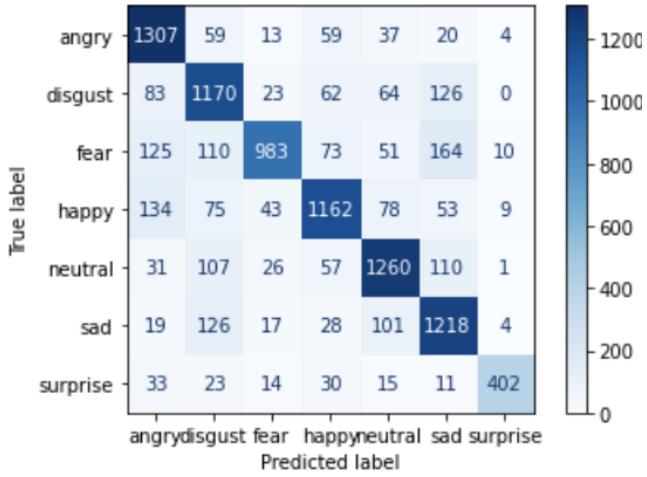

Figure 6.5: confusion matrix visualization Random Forest model with default parameter

labels accounted for 80.50 percent, surprise emotion correct 402 labels accounted for 76.13 percent, The correct sum of 7502 labels gives an accuracy of 77.10 percent.

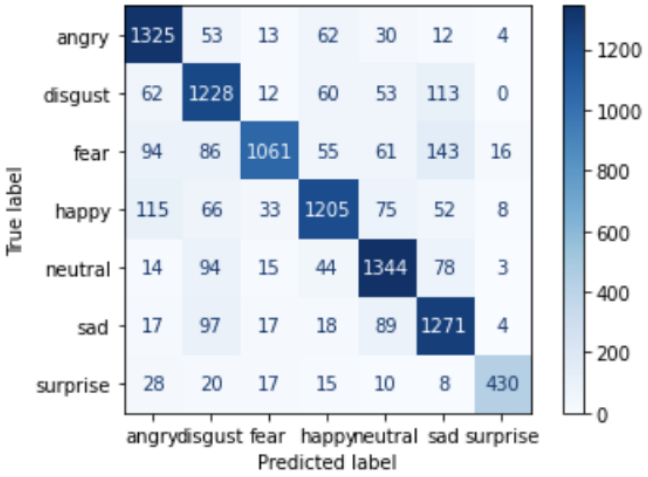

Figure 6.6: confusion matrix visualization Random Forest model with hyperpatameter

While figure 6.6 represents a Random Forest model in hyperparameters that can predict angry emotion correctly, 1325 labels accounted for 88.39 percent, disgust emotion correct 1228 labels accounted for 80.36 percent, fear emotion correct 1061 labels accounted for 69.98 percent, happy emotion correct 1205 labels accounted for 77.54 percent, neutral emotion correct 1344 labels accounted for 84.42 percent, sad emotion correct 1271 labels accounted for 84.00 percent, surprise emotion correct 430 labels accounted for 81.43 percent, The correct total number is 7864 labels, giving an accuracy of 80.82 percent.

6.1.4 K-Nearest Neighbors

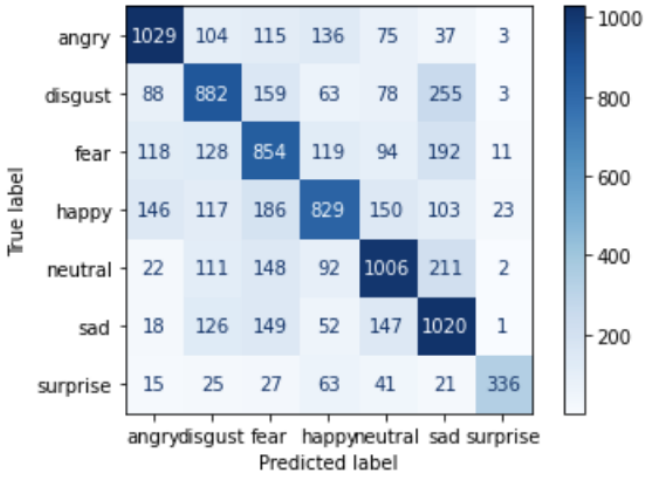

Figure 6.7: confusion matrix visualization K-Nearest Neighbors model with default parameter

Figure 6.7 shows the confusion matrix in the K-Nearest Neighbors model with default parameters that can be predicted angry emotion correct 1029 labels accounted for 68.64 percent, disgust emotion correct 882 labels accounted for 57.72 percent, fear emotion correct 854 labels accounted for 56.33 percent, happy emotion correct 829 labels accounted for 53.34 percent, neutral emotion correct 1006 labels accounted for 63.19 percent, sad emotion correct 1020 labels accounted for 67.41 percent, surprise emotion correct 336 labels accounted for 63.63 percent, The correct sum of 5956 labels gives an accuracy of 61.21 percent.

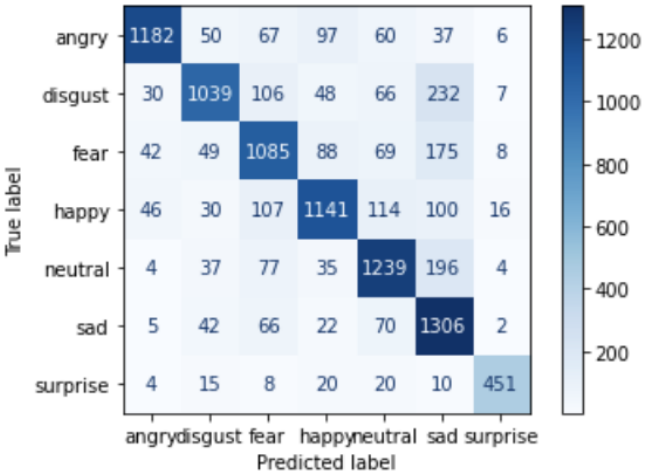

Figure 6.8: confusion matrix visualization K-Nearest Neighbors model with hyperpatameter

While figure 6.8 represents a K-Nearest Neighbors model in hyperparameters that can

predict angry emotion correctly, 1182 labels accounted for 78.85 percent, disgust emotion correct 1039 labels accounted for 67.99 percent, fear emotion correct 1085 labels accounted for 71.56 percent, happy emotion correct 1141 labels accounted for 73.42 percent, neutral emotion correct 1239 labels accounted for 77.82 percent, sad emotion correct 1306 labels accounted for 86.31 percent, surprise emotion correct 451 labels accounted for 85.41 percent, The correct total number is 7443 labels, giving an accuracy of 76.49 percent.

6.1.5 support vector machine

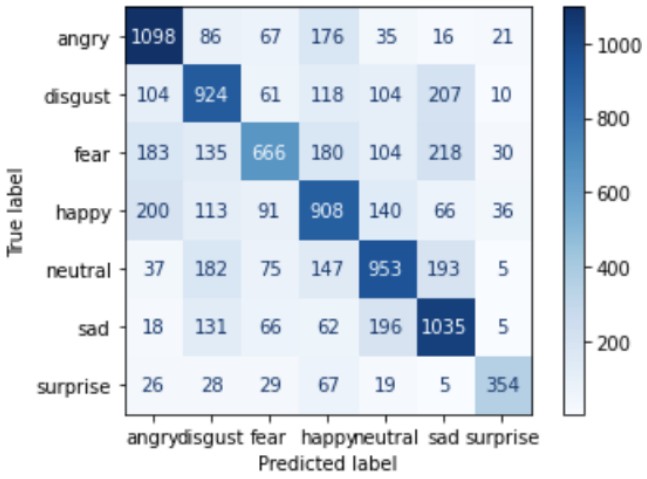

Figure 6.9: confusion matrix visualization support vector machine model

Figure 6.9 shows the confusion matrix in the support vector machine model with default parameters and hyperparameter that can be predicted angry emotion correct 1098 labels accounted for 73.24 percent, disgust emotion correct 924 labels accounted for 60.47 percent, fear emotion correct 666 labels accounted for 43.93 percent, happy emotion correct 908 labels accounted for 58.42 percent, neutral emotion correct 953 labels accounted for 59.86 percent, sad emotion correct 1035 labels accounted for 68.40 percent, surprise emotion correct 354 labels accounted for 67.04 percent, The correct sum of 5938 labels gives an accuracy of 61.02 percent.

6.1.6 Gaussian Naive Bayes

Figure 6.10 shows the confusion matrix in the Gaussian Naive Bayes model with default parameters and hyperparameter that can be predicted angry emotion correct 76 labels ac-

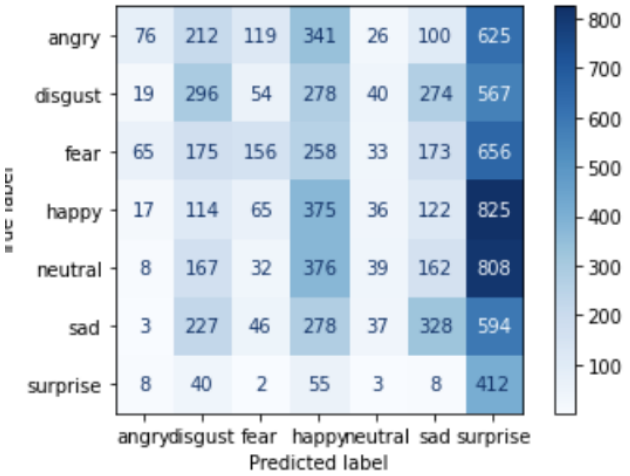

Figure 6.10: confusion matrix visualization Gaussian Naive Bayes model

counted for 5.07 percent, disgust emotion correct 296 labels accounted for 19.37 percent, fear emotion correct 156 labels accounted for 10.29 percent, happy emotion correct 375 labels accounted for 24.13 percent, neutral emotion correct 39 labels accounted for 2.44 percent, sad emotion correct 328 labels accounted for 21.67 percent, surprise emotion correct 412 labels accounted for 78.03 percent, The correct sum of 1682 labels gives an accuracy of 17.28 percent.

6.1.7 eXtreme Gradient Boosting Classification

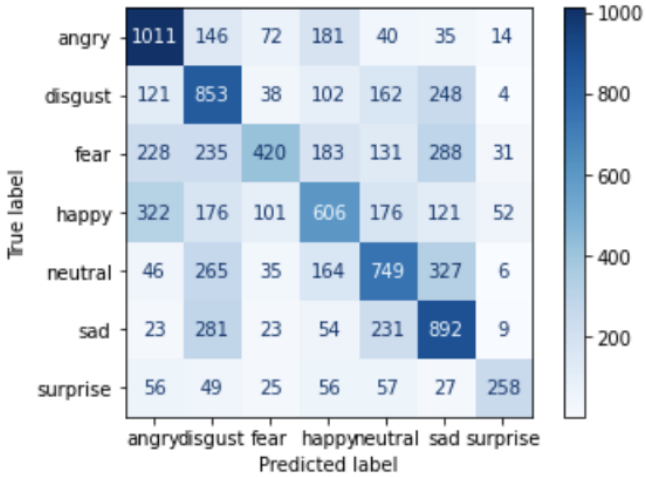

Figure 6.11: confusion matrix visualization eXtreme Gradient Boosting model

Figure 6.11 shows the confusion matrix in the eXtreme Gradient Boosting model with default parameters and hyperparameter that can be predicted angry emotion correct 1011 labels accounted for 67.44 percent, disgust emotion correct 853 labels accounted for 55.82

percent, fear emotion correct 420 labels accounted for 27.70 percent, happy emotion correct 606 labels accounted for 38.99 percent, neutral emotion correct 749 labels accounted for 47.04 percent, sad emotion correct 892 labels accounted for 58.95 percent, surprise emotion correct 258 labels accounted for 48.86 percent, The correct sum of 4789 labels gives an accuracy of 49.21 percent.

6.1.8 Stochastic Gradient Descent

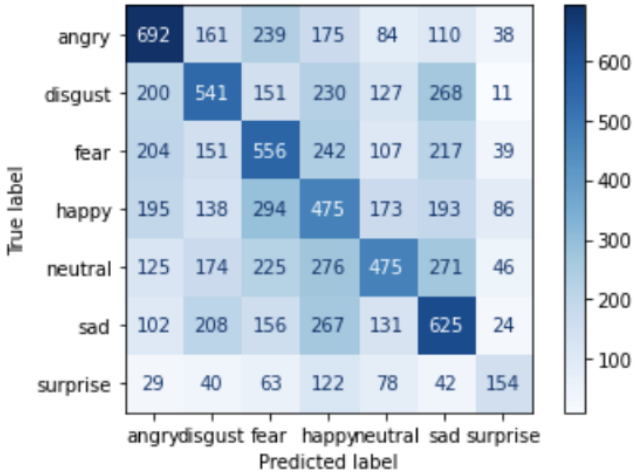

Figure 6.12: confusion matrix visualization Stochastic Gradient Descent model with default parameter

Figure 6.12 shows the confusion matrix in the Stochastic Gradient Descent model with default parameters that can be predicted angry emotion correct 692 labels accounted for 46.16 percent, disgust emotion correct 541 labels accounted for 35.40 percent, fear emotion correct 556 labels accounted for 36.67 percent, happy emotion correct 475 labels accounted for 30.56 percent, neutral emotion correct 475 labels accounted for 29.83 percent, sad emotion correct 625 labels accounted for 41.30 percent, surprise emotion correct 154 labels accounted for 29.16 percent, The correct sum of 3518 labels gives an accuracy of 36.15 percent.

While figure 6.13 represents a Stochastic Gradient Descent model in hyperparameters that can predict angry emotion correct 647 labels accounted for 43.16 percent, disgust emotion correct 578 labels accounted for 37.82 percent, fear emotion correct 471 labels accounted for 31.06 percent, happy emotion correct 472 labels accounted for 30.37 percent, neutral emotion correct 570 labels accounted for 35.80 percent, sad emotion correct 734 labels accounted for

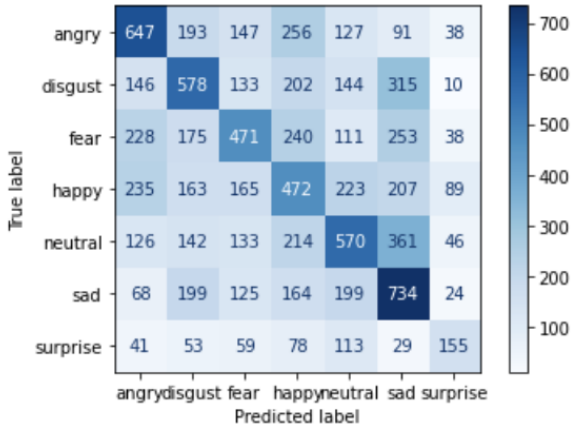

Figure 6.13: confusion matrix visualization Stochastic Gradient Descent model with hyper-patameter

48.51 percent, surprise emotion correct 155 labels accounted for 29.35 percent, The correct total number is 3627 labels, giving an accuracy of 37.27 percent.

6.1.9 Multi-layer Perceptron

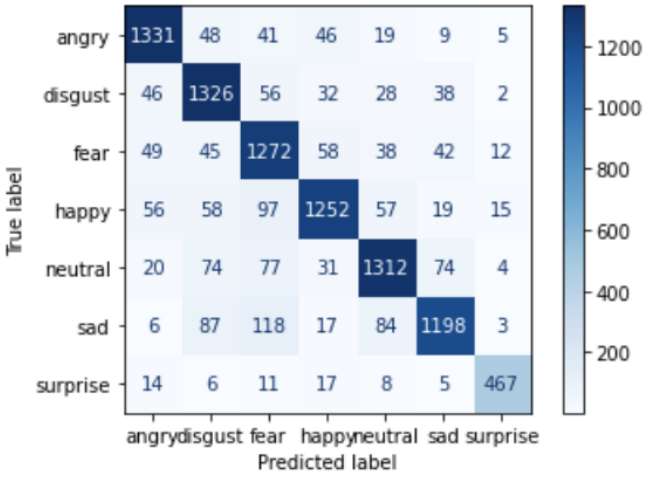

Figure 6.14: confusion matrix visualization Multi-layer Perceptron model with default parameter

Figure 6.14 shows the confusion matrix in the Multi-layer Perceptron model with default parameters that can be predicted angry emotion correct 1331 labels accounted for 88.79 percent, disgust emotion correct 1326 labels accounted for 86.78 percent, fear emotion correct 1272 labels accounted for 83.90 percent, happy emotion correct 1252 labels accounted for 80.56 percent, neutral emotion correct 1312 labels accounted for 82.41 percent, sad emotion

correct 1198 labels accounted for 79.18 percent, surprise emotion correct 467 labels accounted for 88.44 percent, The correct sum of 8158 labels gives an accuracy of 83.84 percent.

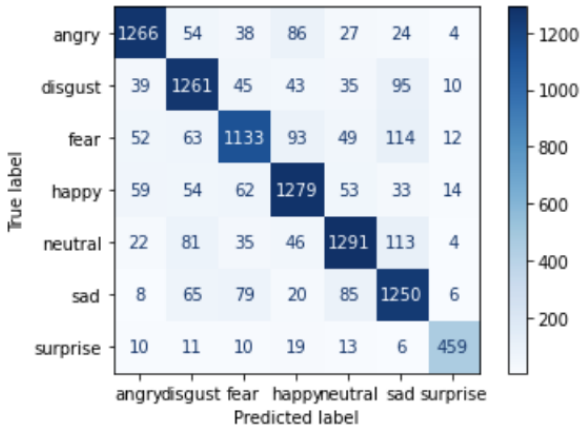

Figure 6.15: confusion matrix visualization Multi-layer Perceptron model with hyperpatameter

While figure 6.15 represents a Multi-layer Perceptron model in hyperparameters that can predict angry emotion correctly, 1266 labels accounted for 84.45 percent, disgust emotion correct 1261 labels accounted for 82.52 percent, fear emotion correct 1133 labels accounted for 74.73 percent, happy emotion correct 1279 labels accounted for 82.30 percent, neutral emotion correct 1291 labels accounted for 81.09 percent, sad emotion correct 1250 labels accounted for 82.61 percent, surprise emotion correct 459 labels accounted for 86.93 percent, The correct total number is 7939 labels, giving an accuracy of 81.59 percent.

6.1.10 Convolutional neural network

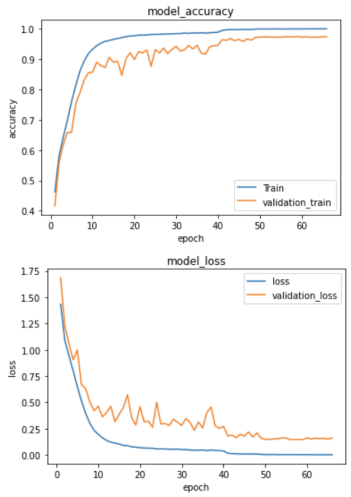

Figure 6.16: Visualization of Loss Functions in Convolutional neural network model

Figure 6.16 is a Convolutional neural network model showing the Learning Curve in the categorical cross-entropy. It can be seen that the validation train and the validation loss fluctuate relatively less. In addition, the train and the validation train are slightly apart, just as the loss and validation loss are apart a little. Therefore, it can be concluded that this model is closer to the term Good Fit Learning Curve than Overfit Learning Curve, which can be applied in real situations.

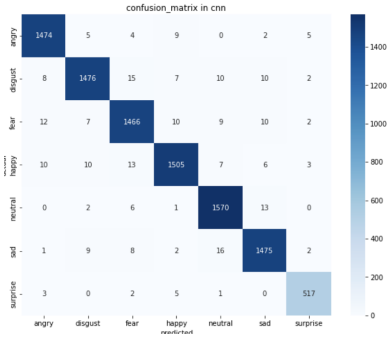

Figure 6.17: confusion matrix visualization Convolutional neural networks model

Figure 6.17 shows the confusion matrix in the Convolutional neural networks model that can be predicted angry emotion correct 1474 labels accounted for 98.33 percent, disgust emotion correct 1476 labels accounted for 96.59 percent, fear emotion correct 1466 labels accounted for 96.70 percent, happy emotion correct 1505 labels accounted for 96.84 percent, neutral emotion correct 1570 labels accounted for 98.61 percent, sad emotion correct 1475

labels accounted for 97.48 percent, surprise emotion correct 517 labels accounted for 97.91 percent, The correct sum of 9483 labels gives an accuracy of 97.44 percent.

6.1.11 Long short-term memory

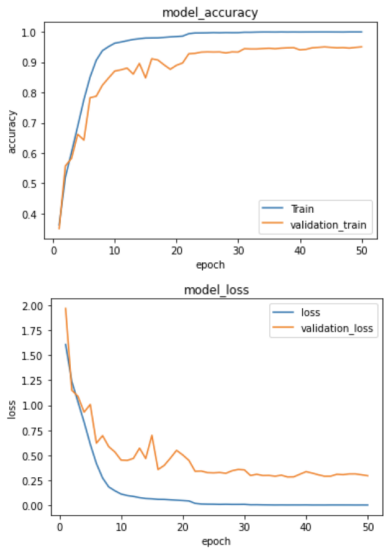

Figure 6.18: Visualization of Loss Functions in Long short-term memory model

Figure 6.18 is a Long short-term memory model showing the Learning Curve in the categorical cross-entropy. It can be seen that the validation train and the validation loss fluctuate relatively less. In addition, the train and the validation train are slightly apart, just as the loss and validation loss are apart a little. Herefore, this model is closer to the term Good Fit Learning Curve than Overfit Learning Curve, which can be applied in real situations.

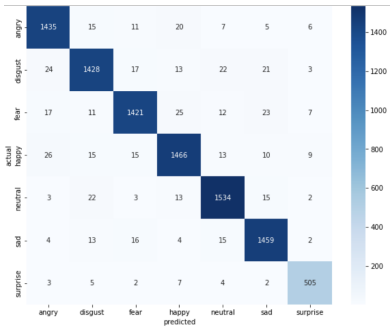

Figure 6.19: confusion matrix visualization Long short-term memory model

Figure 6.19 shows the confusion matrix in the Long short-term memory model that can be predicted angry emotion correct 1435 labels accounted for 95.73 percent, disgust emotion correct 1428 labels accounted for 93.45 percent, fear emotion correct 1421 labels accounted for 93.73 percent, happy emotion correct 1466 labels accounted for 94.33 percent, neutral emotion correct 1534 labels accounted for 96.35 percent, sad emotion correct 1459 labels accounted for

96.43 percent, surprise emotion correct 505 labels accounted for 95.64 percent, The correct sum of 9248 labels gives an accuracy of 94.99 percent.

6.1.12 Gated Recurrent Unit

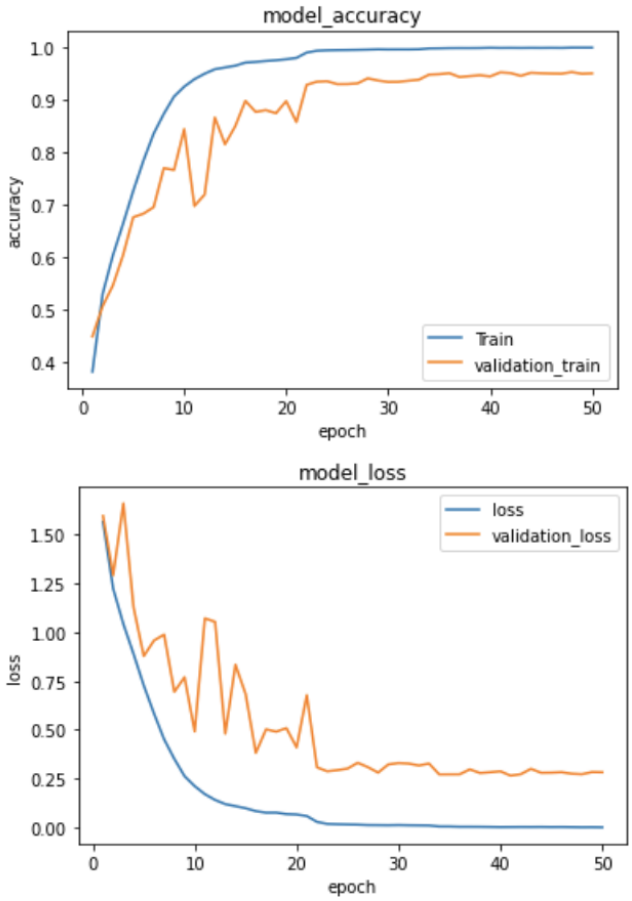

Figure 6.20: Visualization of Loss Functions in Gated Recurrent Unit model

From the figure 6.20 showing the Gated Recurrent Unit model image, it can be seen that the Learning Curve in the categorical cross-entropy in the validation train and validation loss, although highly volatile in the early epochs. However, the volatility is significantly less after many cycles when the epoch is run over many cycles. There is relatively little volatility later. In addition, the train and the validation train are slightly apart, just as the loss and validation loss are slightly apart. Herefore, this model is a Good Fit Learning Curve and not an Overfit Learning Curve, from which this model can be applied in real situations.

Figure 6.21 shows the confusion matrix in the Gated Recurrent Unit model that can be predicted angry emotion correct 1445 labels accounted for 96.39 percent, disgust emotion correct 1436 labels accounted for 93.97 percent, fear emotion correct 1403 labels accounted for 92.54 percent, happy emotion correct 1477 labels accounted for 95.04 percent, neutral emotion correct 1515 labels accounted for 95.16 percent, sad emotion correct 1469 labels accounted for

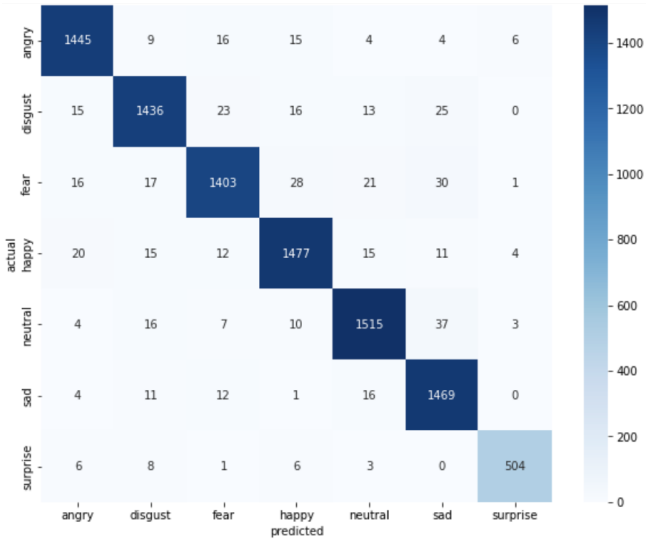

Figure 6.21: confusion matrix visualization Gated Recurrent Unit model

97.09 percent, surprise emotion correct 504 labels accounted for 95.45 percent, The correct sum of 9249 labels gives an accuracy of 95.03 percent.

6.1.13 Deep neural network

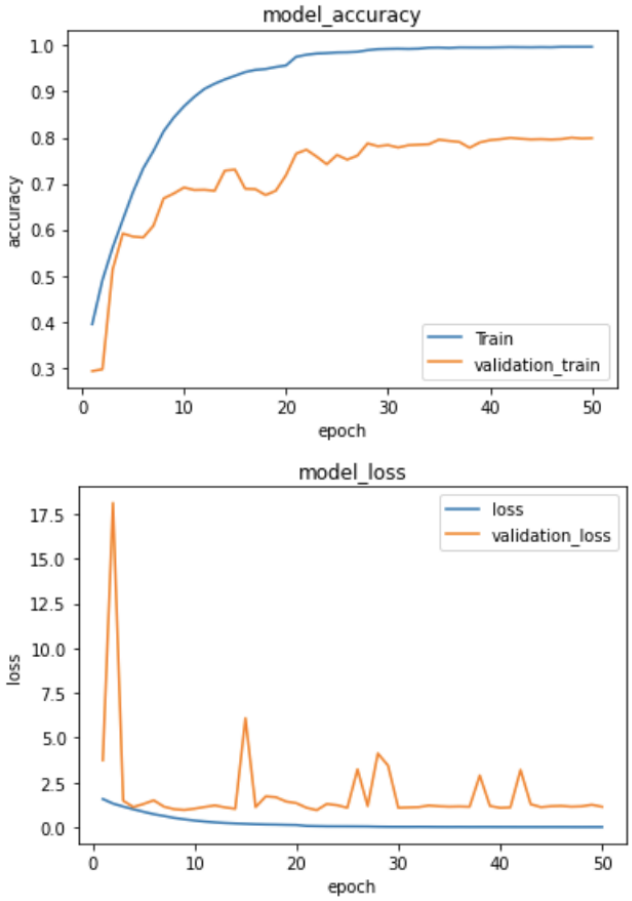

Figure 6.22: Visualization of Loss Functions in Deep neural network model

Model Deep neural network From Figure 6.22. Showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and the validation loss fluctuate considerably. In addition, the train and the validation train are far apart even though the loss and validation loss are not very far apart. However, it can be concluded that this model is not a Good Fit Learning Curve but an Overfit Learning Curve, which cannot be applied in real situations.

Figure 6.23 shows the confusion matrix in the Deep neural network model that can be predicted angry emotion correct 1262 labels accounted for 84.18 percent, disgust emotion correct 1199 labels accounted for 78.46 percent, fear emotion correct 1087 labels accounted for 71.70 percent, happy emotion correct 1220 labels accounted for 78.50 percent, neutral emotion correct 1333 labels accounted for 83.73 percent, sad emotion correct 1216 labels accounted for 80.37 percent, surprise emotion correct 454 labels accounted for 85.98 percent, The correct

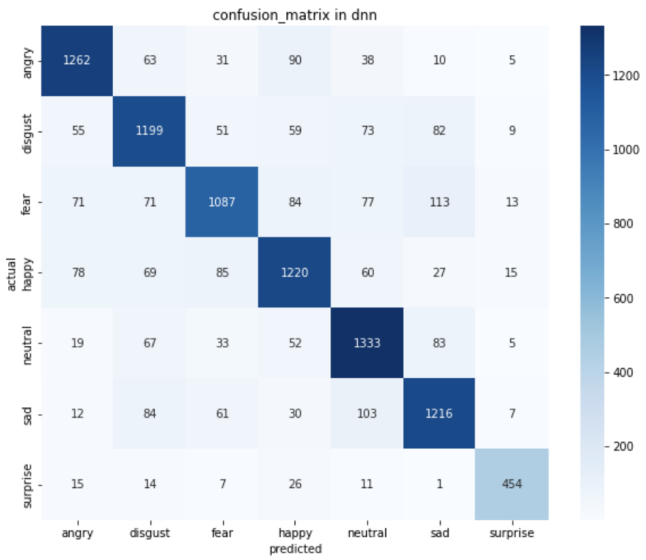

Figure 6.23: confusion matrix visualization Deep neural network model

sum of 7771 labels gives an accuracy of 79.3 percent.

6.1.14 Artificial Neuron Network

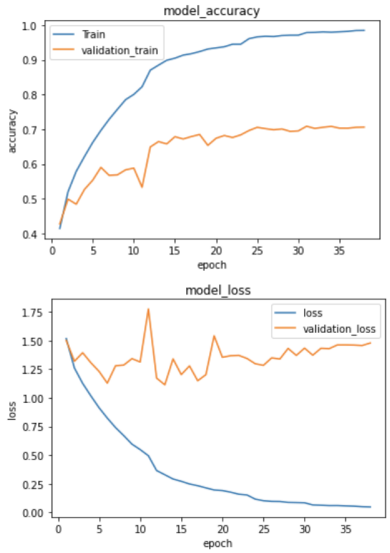

Figure 6.24: Visualization of Loss Functions in Artificial Neuron Network model

From the figure 6.24 of the model Artificial Neuron Network showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and the validation loss fluctuate considerably. In addition, the train and the validation train are far apart, just as the loss and validation loss are so much apart. This model is not a Good Fit Learning Curve but an Overfit Learning Curve, which cannot be applied in real situations.

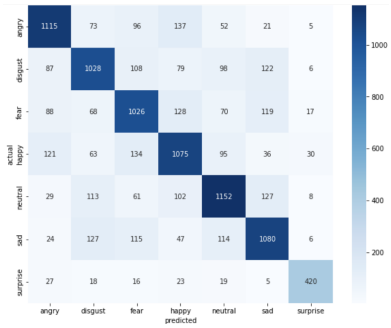

Figure 6.25: confusion matrix visualization Artificial Neuron Network model

Figure 6.25 shows the confusion matrix in the Artificial Neuron Network model that can be predicted angry emotion correct 1115 labels accounted for 74.38 percent, disgust emotion correct 1028 labels accounted for 67.27 percent, fear emotion correct 1026 labels accounted for 67.67 percent, happy emotion correct 1075 labels accounted for 69.17 percent, neutral emotion correct 1152 labels accounted for 72.36 percent, sad emotion correct 1080 labels accounted for

71.38 percent, surprise emotion correct 420 labels accounted for 79.54 percent, The correct sum of 6896 labels gives an accuracy of 69.85 percent.

6.1.15 Convolutional neural network combine with Long short-term memory

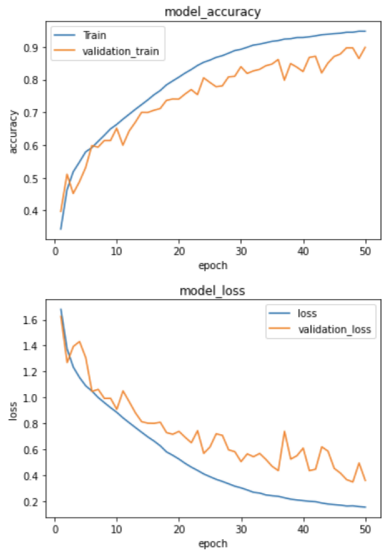

Figure 6.26: Visualization of Loss Functions in Convolutional neural network combine with Long short-term memory models

Figure 6.26 of the CNN combined with the LSTM model shows the Learning Curve in the categorical cross-entropy. It can be seen that the validation train and validation loss fluctuate over time. However, train and validation train are slightly apart, just as loss and validation loss are slightly apart. However, it can be concluded that this model is the Overfit Learning Curve due to the fluctuation of the curve all the time, from which it can be concluded that this model cannot be applied in real situations.

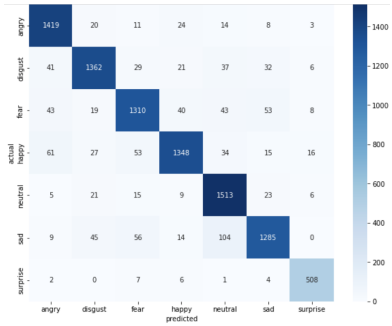

Figure 6.27: confusion matrix visualization Convolutional neural network combine with Long short-term memory model

Figure 6.27 shows the confusion matrix in the Convolutional neural network combined

with a Long short-term memory model that can be predicted angry emotion correct 1419 labels accounted for 94.66 percent, disgust emotion correct 1362 labels accounted for 89.13 percent, fear emotion correct 1310 labels accounted for 86.41 percent, happy emotion correct 1348 labels accounted for 86.74 percent, neutral emotion correct 1513 labels accounted for 95.03 percent, sad emotion correct 1285 labels accounted for 84.93 percent, surprise emotion correct 508 labels accounted for 96.21 percent, The correct sum of 87.45 labels gives an accuracy of 89.39 percent.

6.1.16 Ensemble Learning with hard voting

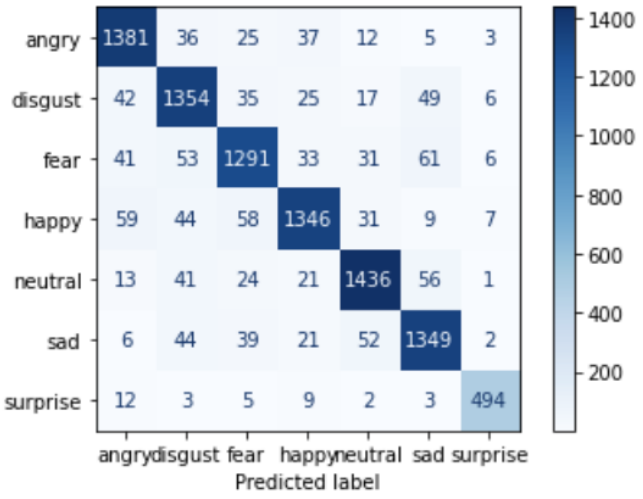

Figure 6.28: confusion matrix visualization Ensemble Learning with hard voting model

Figure 6.28 shows the confusion matrix in the Ensemble Learning with Hard Voting model that can be predicted angry emotion correct 1381 labels accounted for 92.12 percent, disgust emotion correct 1354 labels accounted for 88.61 percent, fear emotion correct 1291 labels accounted for 85.15 percent, happy emotion correct 1346 labels accounted for 86.61 percent, neutral emotion correct 1436 labels accounted for 90.20 percent, sad emotion correct 1349 labels accounted for 89.16 percent, surprise emotion correct 494 labels accounted for 93.56 percent, The correct sum of 8651 labels gives an accuracy of 88.91 percent.

6.1.17 Ensemble Learning with soft voting

Figure 6.29 shows the confusion matrix in the Ensemble Learning with Soft Voting model that can be predicted angry emotion correct 1395 labels accounted for 93.06 percent, disgust emotion correct 1348 labels accounted for 88.21 percent, fear emotion correct 1287 labels accounted for 84.89 percent, happy emotion correct 1371 labels accounted for 88.22 percent, neutral emotion correct 1439 labels accounted for 90.38 percent, sad emotion correct 1395 labels accounted for 92.20 percent, surprise emotion correct 505 labels accounted for 95.64 percent, The correct sum of 8740 labels gives an accuracy of 89.82 percent.

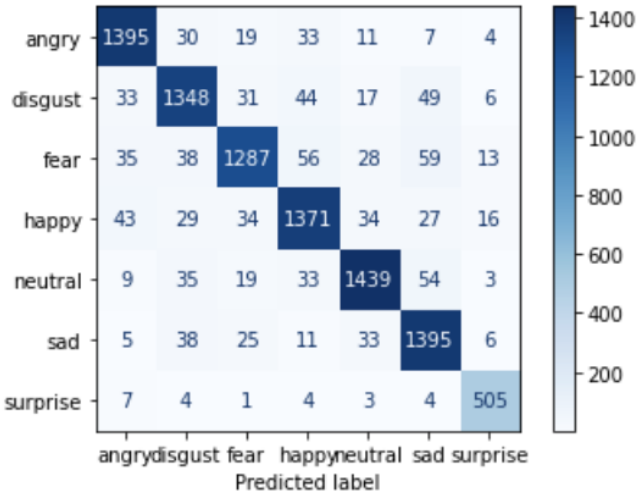

Figure 6.29: confusion matrix visualization Ensemble Learning with soft voting model

6.2 predict gender and emotion labels

6.2.1 Decision Tree

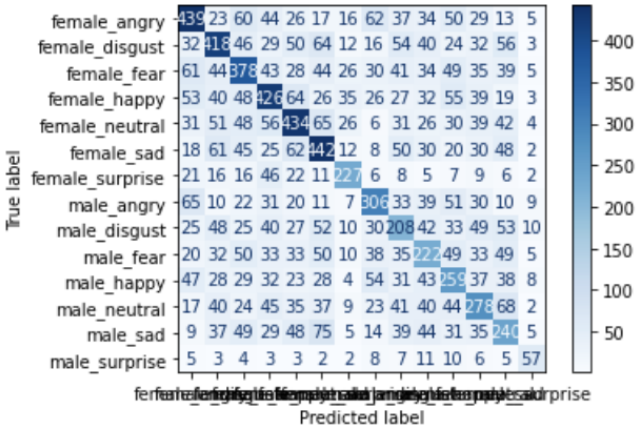

Figure 6.30: confusion matrix visualization Decision Tree model with default parameter

From the figure 6.30 that predicts the Decision Tree model in the default parameter, we can clearly see that the Female has the correct number of 2764 labels representing 49.13 percent. While male has the correct number of 1570 labels, it is 38.24 percent. Therefore, this model has the correct number of 4334 labels. It gives a total accuracy equal to 44.54 percent.

In the hyperparameter, figure 6.31 will show that the Female has the correct number 2416 label accounting for 42.95 percent. At the same time, the male has the correct number of 1298 labels. It is 31.61 percent. Therefore, this model has the correct number of 3714 labels. It gives

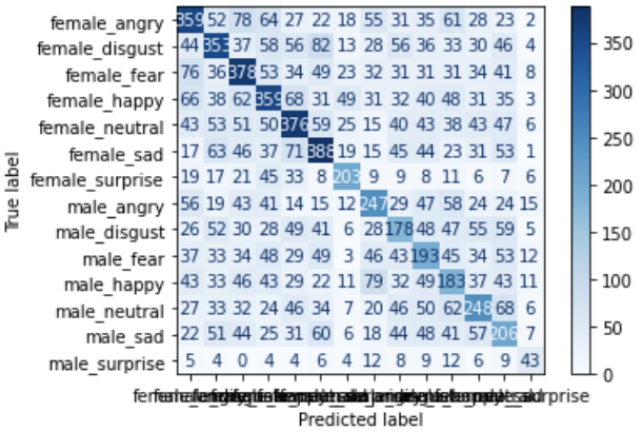

Figure 6.31: confusion matrix visualization Decision Tre model with hyperpatameter

a total accuracy equal to 38.17 percent.

6.2.2 Logistic Regression

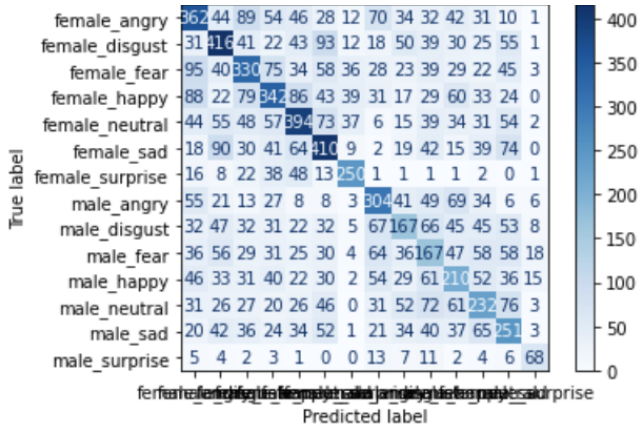

Figure 6.32: confusion matrix visualization logistic regression model with default parameter

From the figure 6.32 that predicts the Logistic Regression model in the default parameter, we can see that the Female has the correct number of 2504 labels representing 44.51 percent. While male has the correct number of 1399 labels, it is 34.08 percent. Therefore, this model has the correct number of 3903 labels. It gives a total accuracy equal to 40.11 percent.

In the hyperparameter, figure 6.33 will show that the Female has the correct number 2557 label accounting for 45.45 percent. In contrast, the male has the correct number of 1405 labels. It is 34.22 percent. Therefore, this model has the correct number of 3962 labels. It gives a total accuracy equal to 40.71 percent.

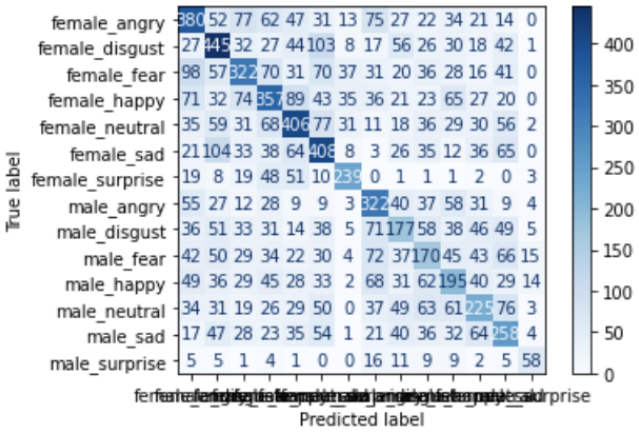

Figure 6.33: confusion matrix visualization logistic regression model with hyperpatameter

6.2.3 Random Forest

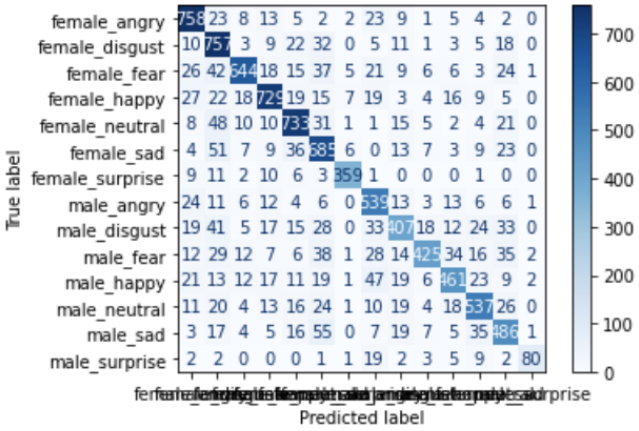

Figure 6.34: confusion matrix visualization random forest model with default parameter

From the figure 6.34 that predicts the Random Forest model in the default parameter, we can see that the Female has the correct number of 4665 labels representing 82.93 percent. While male has the correct number of 2935 labels, it is 71.49 percent. Therefore, this model has the correct number of 7600 labels. It gives a total accuracy equal to 78.10 percent.

In the hyperparameter, figure 6.35 will show that the Female has the correct number 4975 label accounting for 88.44 percent. In comparison, the male has the correct number of 3336 labels. It is 81.26 percent. Therefore, this model has the correct number of 8311 labels. It gives a total accuracy equal to 85.41 percent.

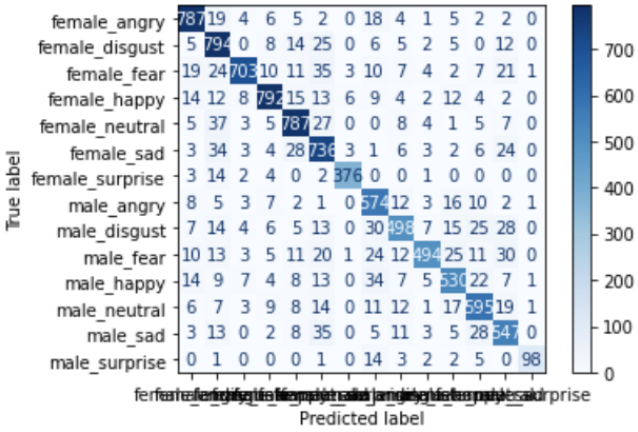

Figure 6.35: confusion matrix visualization random forest model with hyperpatameter

6.2.4 K-Nearest Neighbors

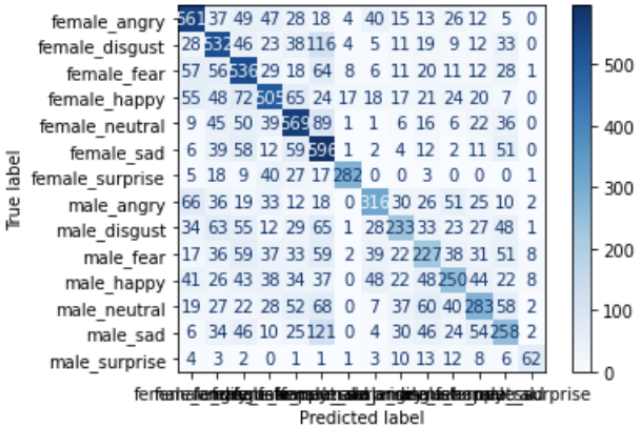

Figure 6.36: confusion matrix visualization K-Nearest Neighbors model with default parameter

From the figure 6.36 that predicts the K-Nearest Neighbors model in the default parameter, we can see that the Female has the correct number of 3581 labels representing 63.66 percent. While male has the correct number of 1629 labels, it is 39.68 percent. Therefore, this model has the correct number of 5210 labels. It gives a total accuracy equal to 53.54 percent.

In the hyperparameter, figure 6.37 will show that the Female has the correct number 4456 label accounting for 79.21 percent. At the same time, the male has the correct number of 2796 labels. It is 68.11 percent. Therefore, this model has the correct number of 7252 labels. It gives a total accuracy equal to 74.53 percent.

Figure 6.37: confusion matrix visualization K-Nearest Neighbors model with hyperpatameter

6.2.5 support vector machine

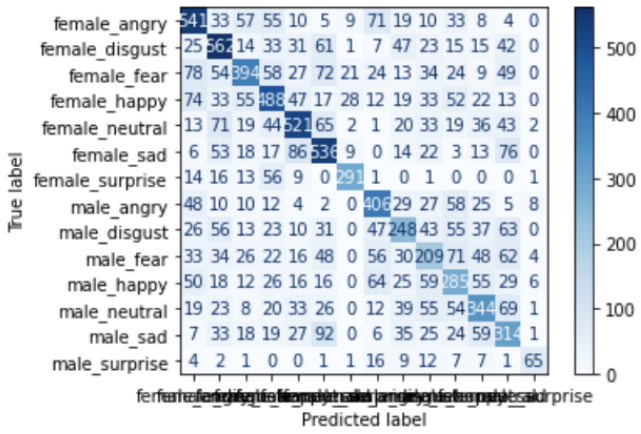

Figure 6.38: confusion matrix visualization support vector machine model with default parameter

From the figure 6.38 that predicts the support vector machine model in the default parameter, we can see that the Female has the correct number of 3333 labels representing 59.25 percent. While male has the correct number of 1871 labels, it is 45.57 percent. Therefore, this model has the correct number of 5204 labels. It gives a total accuracy equal to 53.48 percent.

In the hyperparameter, figure 6.39 will show that the Female has the correct number 3318 label accounting for 58.98 percent. In comparison, the male has the correct number of 1885 labels. It is 45.91 percent. Therefore, this model has the correct number of 5203 labels. It gives a total accuracy equal to 53.47 percent.

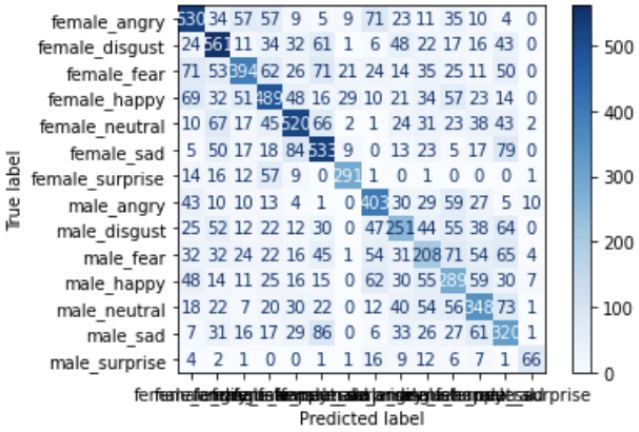

Figure 6.39: confusion matrix visualization support vector machine model with hyperpatameter

6.2.6 Gaussian Naive Bayes

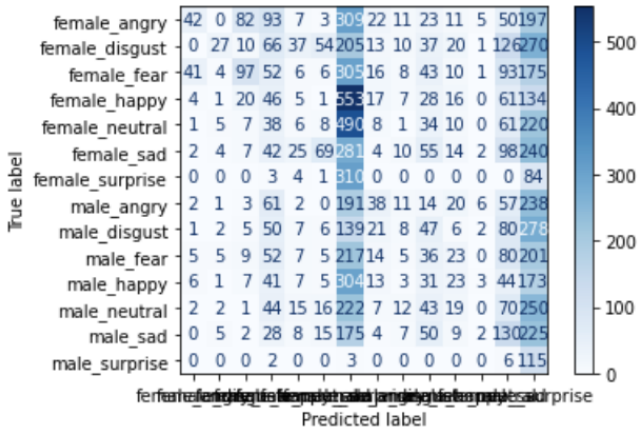

Figure 6.40: confusion matrix visualization Gaussian Naive Bayes model with default parameter

From the figure 6.40 that predicts the Gaussian Naive Bayes model in the default parameter and hyperparameter, we can see that the Female has the correct number of 597 labels representing 10.61 percent. While male has the correct number of 350 labels, it is 8.52 percent. Therefore, this model has the correct number of 947 labels. It gives a total accuracy equal to 9.73 percent.

6.2.7 eXtreme Gradient Boosting

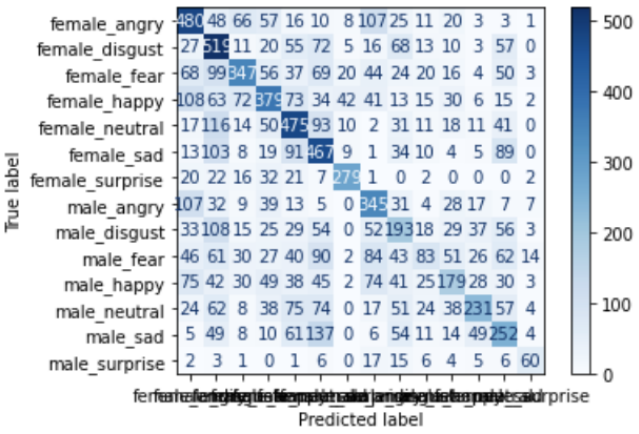

Figure 6.41: confusion matrix visualization eXtreme Gradient Boosting model with default parameter

From the figure 6.41 that predicts the eXtreme Gradient Boosting model in the default parameter and hyperparameter, we can see that the Female has the correct number of 2946 labels representing 52.37 percent. While male has the correct number of 1343 labels, it is 32.71 percent. Therefore, this model has the correct number of 4289 labels. It gives a total accuracy equal to 44.08 percent.

6.2.8 Stochastic Gradient Descent

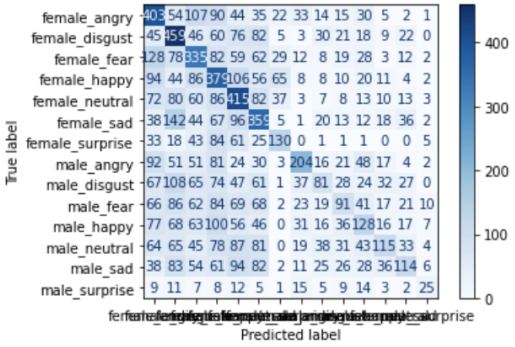

Figure 6.42: confusion matrix visualization stochastic gradient descent model with default parameter

From the figure 6.42 that predicts the stochastic gradient descent model in the default parameter, we can see that the Female has the correct number of 2408 labels representing 44.08 percent. While male has the correct number of 758 labels, it is 18.46 percent. Therefore, this model has the correct number of 3238 labels. It gives a total accuracy equal to 33.27 percent.

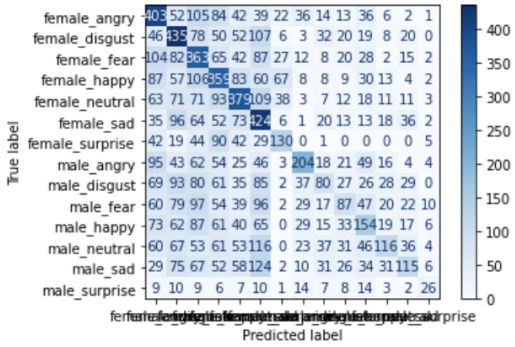

Figure 6.43: confusion matrix visualization stochastic gradient descent model with hyper-patameter

In the hyperparameter, figure 6.43 will show that the Female has the correct number 2493 label accounting for 44.32 percent. At the same time, male has the correct number of 782 labels. It is 19.04 percent. Therefore, this model has the correct number of 3275 labels. It gives a total accuracy equal to 33.65 percent.

6.2.9 Multi-layer Perceptron

Figure 6.44: confusion matrix visualization Multi-layer Perceptron model with default parameter

From the figure 6.44 that predicts the model Multi-layer Perceptron in the default parameter, we can see that Female has the correct number of 4603 labels representing 81.83 percent. While male has the correct number of 3055 labels, it is 74.42 percent. Therefore, this model has the correct number of 7658 labels. It gives a total accuracy equal to 78.70 percent.

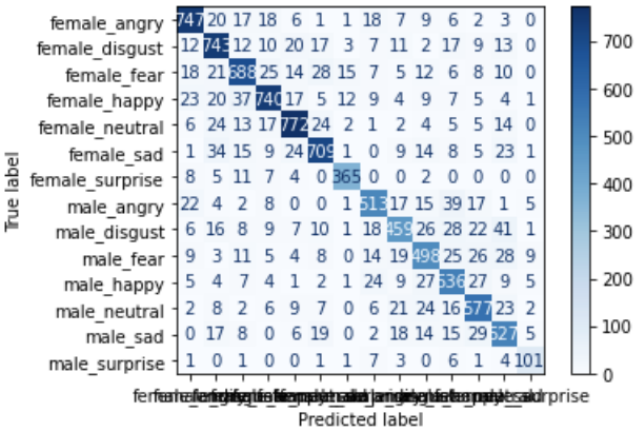

Figure 6.45: confusion matrix visualization Multi-layer Perceptron model with hyperpatameter

In the hyperparameter, figure 6.45 will show that the Female has the correct number 4764 label accounting for 84.69 percent. In contrast, the male has the correct number of 3211 labels. It is 78.22 percent. Therefore, this model has the correct number of 7975 labels. It gives a total accuracy equal to 81.96 percent.

6.2.10 Convolutional neural network

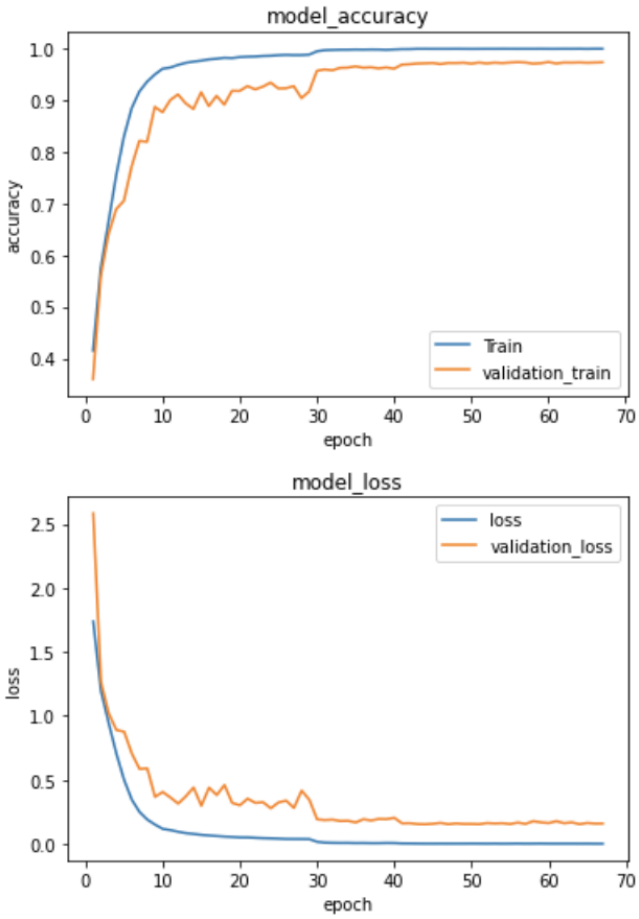

Figure 6.46: Visualization of Loss Functions in Convolutional neural network model

Figure 6.46 of the CNN model shows the Learning Curve in the categorical cross-entropy. It can be seen that the validation train and the validation loss fluctuate relatively less. In addition, the train and the validation train are slightly apart, just as the loss and validation loss are apart a little. Therefore, it can be concluded that this model is closer to the term Good Fit Learning Curve than Overfit Learning Curve, which can be applied in real situations.

From the figure 6.47 predicts the CNN model, we can see that the Female has the correct number of 5531 labels representing 98.32 percent. While male has the correct number of 3948 labels, it is 96.17 percent. Therefore, this model has the correct number of 9479 labels. It gives a total accuracy equal to 97.35 percent.

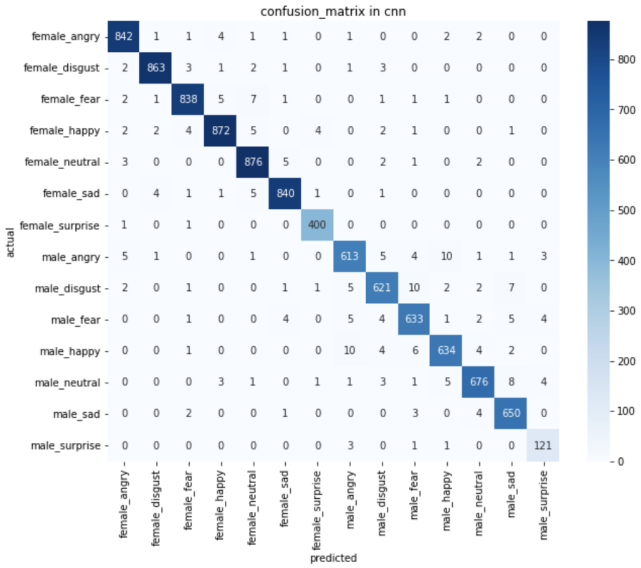

Figure 6.47: confusion matrix visualization Convolutional neural network model

6.2.11 Long short-term memory

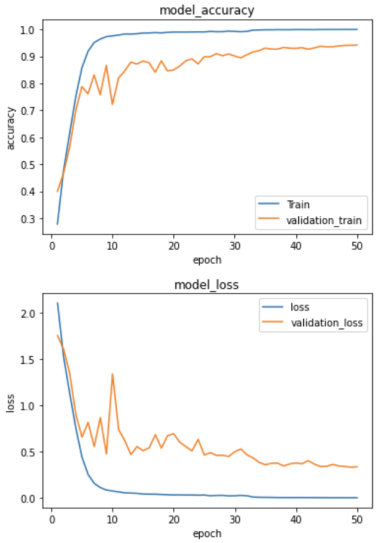

Figure 6.48: Visualization of Loss Functions in Long short-term memory model

From the figure 6.48 of the LSTM model showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and validation loss fluctuate quite a lot at the beginning but at the epoch greater than 15. The model fluctuates very low. Also, the train and the validation train are slightly apart, just as the loss and validation loss are slightly apart. Therefore, it can be concluded that this model is a Good Fit Learning Curve and not an Overfit Learning Curve, from which this model can be applied in real situations.

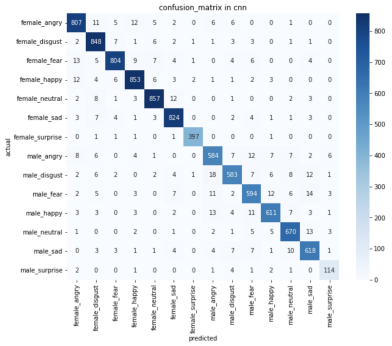

Figure 6.49: confusion matrix visualization Long short-term memory model

From the figure 6.49 predicts the LSTM model, we can see that the Female has the correct number of 5390 labels representing 95.82 percent. While male has the correct number of 3774 labels, it is 91.93 percent. Therefore, this model has the correct number of 9164 labels. It gives a total accuracy equal to 94.00 percent.

6.2.12 Gated Recurrent Unit

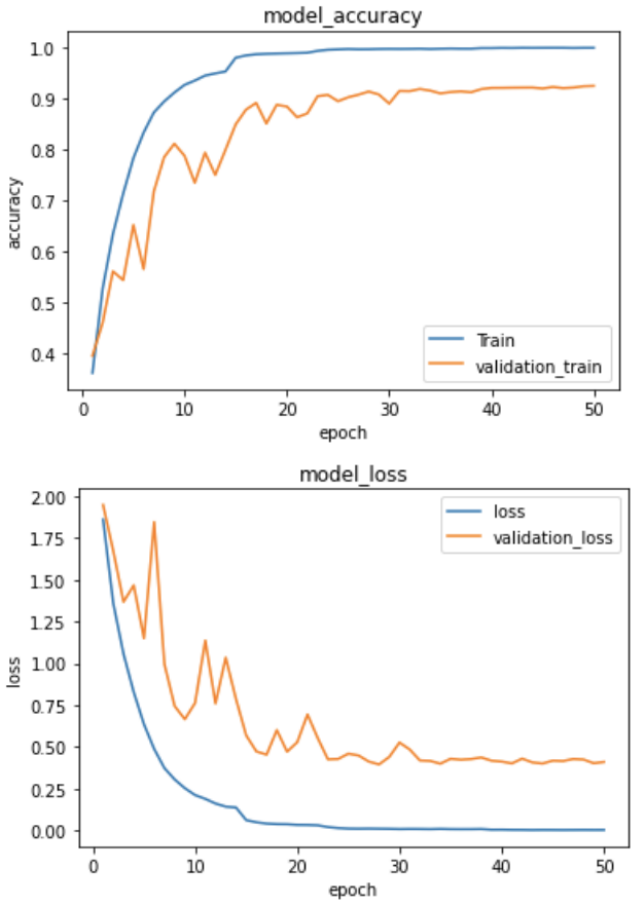

Figure 6.50: Visualization of Loss Functions in Gated Recurrent Unit model

From the figure 6.50 of the GRU model showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and validation loss, although highly volatile in the early epochs. Nevertheless, when the epoch is run more than 30 cycles, the volatility is less, so there is relatively less volatility in the later stages. In addition, the train and the validation train are slightly apart, just as the loss and validation loss are slightly apart. Therefore, it can be concluded that this model is a Good Fit Learning Curve and not an Overfit Learning Curve, from which this model can be applied in real situations.

From the figure 6.51 predicts the GRU model, we can see that the Female has the correct number of 5313 labels representing 94.45 percent. While male has the correct number of 3682 labels, it is 89.69 percent. Therefore, this model has the correct number of 8995 labels. It gives a total accuracy equal to 92.34 percent.

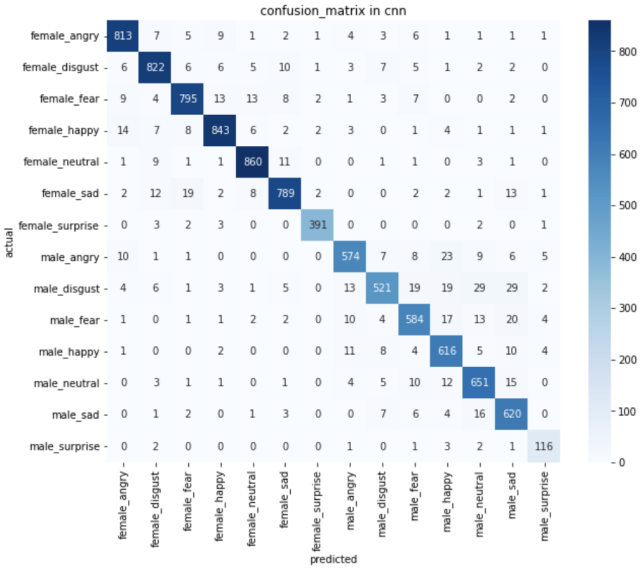

Figure 6.51: confusion matrix visualization Gated Recurrent Unit model

6.2.13 Deep neural network

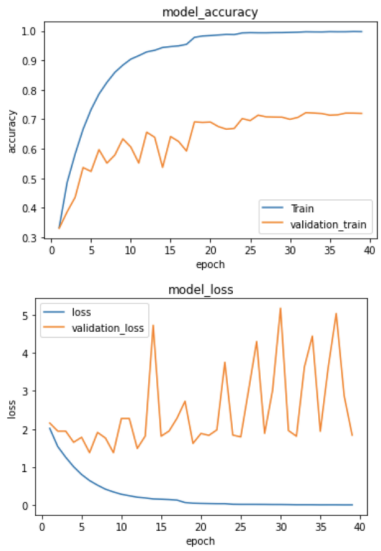

Figure 6.52: Visualization of Loss Functions in Deep neural network model

From the figure 6.52 of the DNN model showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and the validation loss fluctuate so much that the true loss and accuracy are unknown. In addition, trains and validation trains are very far apart, just as loss and accuracy are. Validation loss is far apart. Therefore, it can be concluded that this model is not a Good Fit Learning Curve but an Overfit Learning Curve, which cannot be applied in real situations.

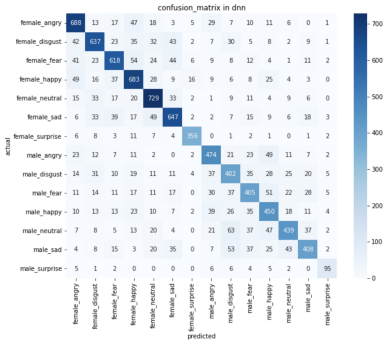

Figure 6.53: confusion matrix visualization Deep neural network model

From the figure 6.53 that predicts the DNN model, we can see that the Female has the correct number of 4358 labels representing 77.47 percent. While male has the correct number of 2673 labels, it is 65.11 percent. Therefore, this model has the correct number of 7031 labels. It gives a total accuracy equal to 71.59 percent.

6.2.14 Artificial Neuron Network

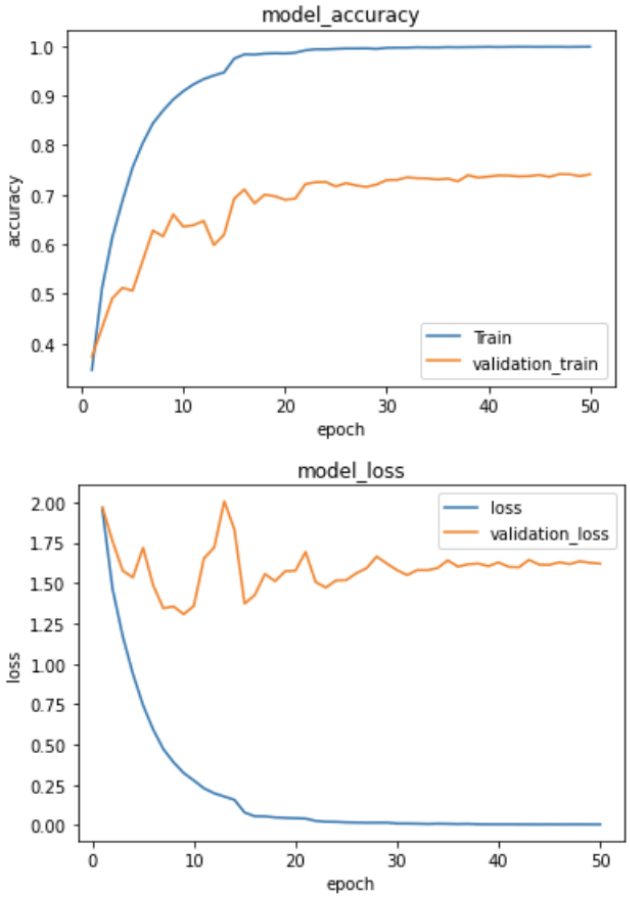

Figure 6.54: Visualization of Loss Functions in Artificial Neuron Network model

From the figure 6.54 of the model ANN showing the Learning Curve in the categorical cross-entropy, it can be seen that the validation train and the validation loss fluctuate considerably. In addition, the train and the validation train are far apart, just as the loss and validation loss are far apart. Therefore, it can be concluded that this model is not a Good Fit Learning Curve but an Overfit Learning Curve, which cannot be applied in real situations.

From the figure 6.55 predicts the ANN model, we can see that the Female has the correct number of 4435 labels representing 78.84 percent. While male has the correct number of 2778 labels, it is 67.67 percent. Therefore, this model has the correct number of 7213 labels. It gives a total accuracy equal to 73.5 percent.

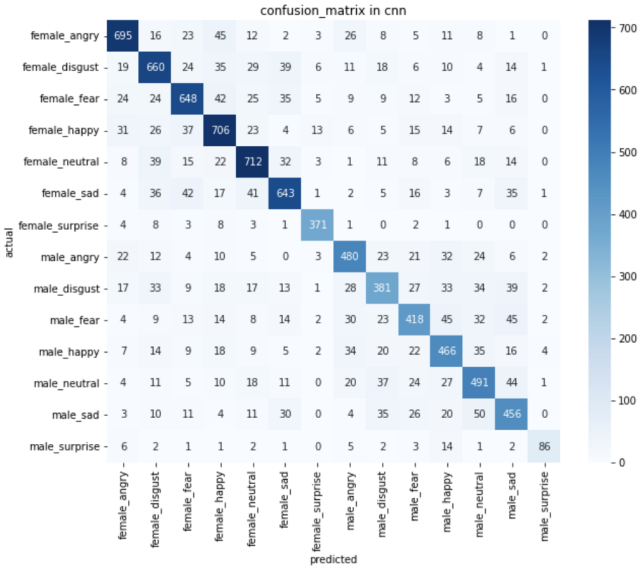

Figure 6.55: confusion matrix visualization Artificial Neuron Network model

6.2.15 Convolutional neural network combine with Long short-term memory

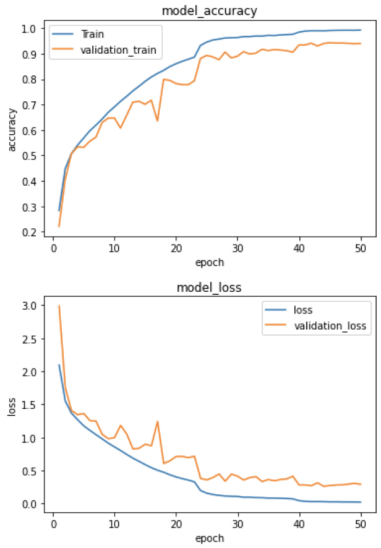

Figure 6.56: Visualization of Loss Functions in Convolutional neural network combine with Long short-term memory model

Figure 6.56 of the CNN combined with the LSTM model shows that the Learning Curve in the categorical cross entropy can be seen that the validation train and validation loss has highly volatile in the early epochs. However, when the epoch is run over 30 cycles, the volatility is less, and there is relatively less volatility in the later stages. In addition, the train

and the validation train are slightly apart, just as the loss and validation loss are slightly apart. Therefore, it can be concluded that this model is a Good Fit Learning Curve and not an Overfit Learning Curve, from which this model can be applied in real situations.

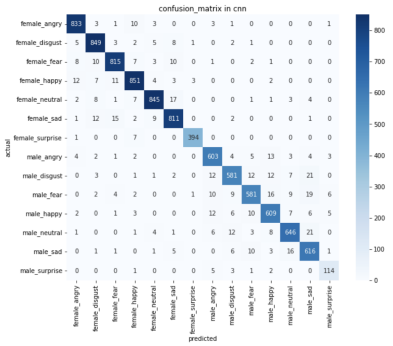

Figure 6.57: confusion matrix visualization Convolutional neural network combine with Long short-term memory model

From the figure 6.57 predicts the CNN combined with the LSTM model, we can see that Female has the correct number of 5398 labels representing 95.96 percent. While male has the correct number of 3750 labels, it is 91.35 percent. Therefore, this model has the correct number of 9148 labels. It gives a total accuracy equal to 93.87 percent.

6.2.16 Ensemble Learning with hard voting

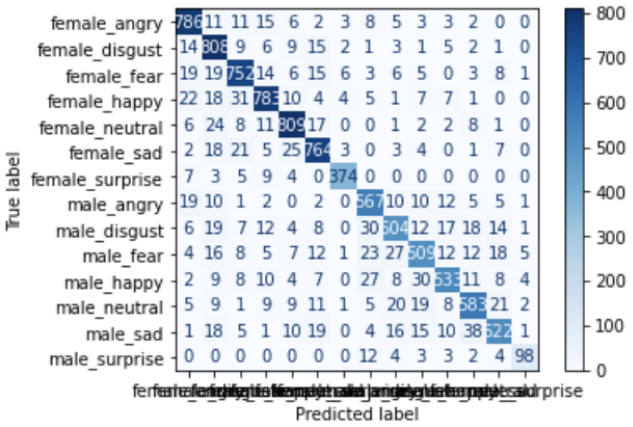

Figure 6.58: confusion matrix visualization Ensemble Learning with hard voting model

From the figure 6.58 predicts the Ensemble Learning with Hard Voting model, we can see that the Female has the correct number of 5076 labels representing 90.24 percent. While male has the correct number of 3316 labels, it is 80.77 percent. Therefore, this model has the correct number of 8392 labels. It gives a total accuracy equal to 86.24 percent.

6.2.17 Ensemble Learning with soft voting

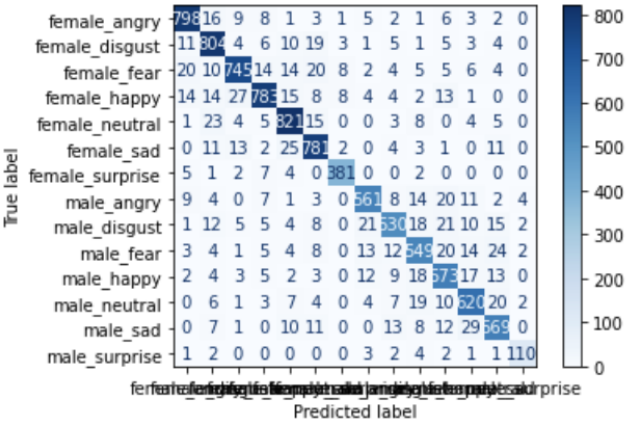

Figure 6.59: confusion matrix visualization Ensemble Learning with soft voting model

From the figure 6.59 predicts the Ensemble Learning with Soft Voting model, we can see that the Female has the correct number of 5113 labels representing 90.89 percent. While male has the correct number of 3521 labels, it is 85.77 percent. Therefore, this model has the correct number of 8634 labels. It gives a total accuracy equal to 88.64 percent.

6.3 Compare model

algorithm	default parameter	hyperparameter	best accuracy
Decision Tree	**49.26**	47.18	49.26
Logistic Regression	**43.36**	42.58	43.36
Random Forest	77.10	**80.82**	80.82
K-Nearest Neighbors	61.21	**76.49**	76.49
support vector machine	**61.02**	**61.02**	61.02
Gaussian Naive Bayes	**17.28**	**17.28**	17.28
XGBoost	**49.21**	**49.21**	49.21
SGD	36.15	**37.27**	37.27
Multi-layer Perceptron	**83.84**	81.59	83.84
average model	53.17	54.83	55.40

Table 6.1: comparative 9 models with only emotion label between default parameter and hyperparameter

algorithm	default parameter	hyperparameter	best accuracy
Decision Tree	**44.54**	38.17	44.54
Logistic Regression	40.11	**40.71**	40.71
Random Forest	78.10	**85.41**	85.41
K-Nearest Neighbors	53.54	**74.53**	74.53
support vector machine	**53.48**	53.47	53.48
Gaussian Naive Bayes	**9.73**	**9.73**	9.73
XGBoost	**44.08**	**44.08**	44.08
SGD	33.27	**33.65**	33.65
Multi-layer Perceptron	78.70	**81.96**	81.96
average model	48.39	51.30	52.01

Table 6.2: comparative 9 models with gender and emotion between labels default parameter and hyperparameter

This research has compared the default parameter and hyperparameter in only emotion in table 6.1 and gender and emotion in table 6.2. There is some model where a hyperparameter is more accurate than the default parameter. However, in some models default parameter is more precise because RandomizedSearchCV For this study, only ten parameters were randomly selected and the best one was determined. This research does not use GridSearchCV because some models take so long to run that they cannot be completed. This sometimes makes the default parameter more accurate than the hyperparameter.

algorithm	angry	disgust	fear	happy	neutral	sad	surprise	accuracy
Decision Tree	55.17	46.66	45.25	45.94	50.37	52.74	47.91	49.26
Logistic Regression	54.16	43.84	34.30	38.93	37.87	49.43	49.43	43.36
Random Forest	88.39	80.36	69.98	77.54	84.42	84.00	81.43	80.82
K-Nearest Neighbors	78.85	67.99	71.56	73.42	77.82	86.31	85.41	76.49
support vector machine	73.24	60.47	43.93	58.42	59.86	68.40	67.04	61.02
Gaussian Naive Bayes	5.07	19.37	10.29	24.13	2.44	21.67	78.03	17.28
XGBoost	67.44	55.82	27.70	38.99	47.04	58.95	48.86	49.21
SGD	43.16	37.82	31.06	30.37	35.80	48.51	29.35	37.27
Multi-layer Perceptron	88.79	86.78	83.90	80.56	82.41	79.18	88.44	83.84
EL with hard vote	92.12	88.61	85.15	86.61	90.20	89.16	93.56	88.91
EL with soft vote	93.06	88.21	84.89	88.22	90.38	92.20	95.64	89.82
CNN	**98.33**	**96.59**	**96.70**	**96.84**	**98.61**	**97.48**	**97.91**	**97.44**
LSTM	95.73	93.45	93.73	94.33	96.35	96.43	95.64	94.99
gated recurrent unit	96.39	93.97	92.54	95.04	95.16	97.09	95.45	95.03
Deep neural network	84.18	78.46	71.70	78.50	83.73	80.37	85.98	79.30
artificial neuron network	74.38	67.27	67.67	69.17	72.36	71.38	79.54	69.85
CNN and LSTM	94.66	89.13	86.41	86.74	95.03	84.93	96.21	89.39
average model	75.47	70.28	64.51	68.45	70.57	74.01	77.40	70.78

Table 6.3: comparative 17 models with gender and emotion labels

The most accurate model is CNN in table 6.3, which only shows emotion. It can be more accurate than any other model in every emotion, with an accuracy of 97.44. percent, while machine learning has been overwhelmingly successful in implementing ensemble learning

with the soft vote that increases machine learning accuracy to 89.82 percent. The table also indicated that the slightest accurate sound was the fear sound, with an average accuracy of 64.51 percent. While the most accurate sound is the surprise sound has an average accuracy of 77.40 percent. In table 6.4 that shows gender and emotion, We can see that the average

algorithm	female	male	accuracy
Decision Tree	49.13	38.24	44.54
Logistic Regression	45.45	34.22	40.71
Random Forest	88.44	81.26	85.41
K-Nearest Neighbors	79.21	68.11	74.53
support vector machine	59.25	45.57	53.48
Gaussian Naive Bayes	10.61	8.52	9.73
XGBoost	52.37	32.71	44.08
SGD	44.32	19.04	33.65
Multi-layer Perceptron	84.69	78.22	81.96
EL with hard	90.24	80.77	86.24
EL with soft	90.89	85.77	88.64
CNN	**98.32**	**96.17**	**97.35**
LSTM	95.82	91.93	94.00
gated recurrent unit	94.45	89.69	92.34
Deep neural network	77.47	65.11	71.59
artificial neuron network	78.84	67.67	73.50
CNN and LSTM	95.96	91.35	93.87
average model	72.67	63.19	68.56

Table 6.4: comparative 17 models with only emotion labels

female voice has an accuracy of 72.67 percent. This is much higher than the average male voice, which has an accuracy of 63.19 percent, and the most accurate model is CNN, with 97.35 percent accuracy. The accuracy of the female voice was 98.32 percent, and the accuracy of the male voice was 96.17 percent. While the most accurate machine learning model was ensemble learning with the soft vote, it had an accuracy of 85.77 percent for males and 90.89 percent for females. Furthermore, the accuracy of the model was 88.64 percent.

CHAPTER 7

Discussion

From experiments comparing the predictions of machine learning models with default parameters and machine learning models with hyperparameters. Machine learning models with hyperparameter. It should be more precise than the default parameter because hyperparameter use tuning parameters to increase accuracy. However, we will notice that even though Table 6.1 compares the default parameter and hyperparameter in only emotion labels with machine learning nine models, Table 6.2 compares the default parameter and hyperparameter in gender and emotion labels with machine learning nine models. The hyperparameters are more accurate than the default parameter in some models, but in some models, The hyperparameter has less accurate accuracy than the default parameter because this research was unable to tune the parameter using GridSearchCV used to find the best parameter out of all parameters because the time spent in tuning parameter with GridSearchCV is so long that it cannot be completed. It can be seen from figure 7.1 and figure 7.2. that shows running model per 1parameter, which is very difficult for some models to complete GridSearchCV because it takes too long. This research, therefore, uses RandomizedSearchCV. This research predicted 17 models. From Table 6.3 comparing only emotion labels and Table 6.4 comparing emotion and gender labels, it can be seen that the model that performs best in machine learning is Ensemble Learning with soft, which succeeded by developing the best voting model of machine learning. In this research, it was observed that Multi-layer Perceptron is the most accurate model of machine learning. Therefore, Ensemble Learning with soft in this research used a Multi-layer Perceptron 5 parameter vote was 89.82 percent predictable

from only emotion label, and 88.64 percent predicted from emotion and label. In contrast, the best deep learning model is the CNN model, which predicts only the emotion label at 97.44 percent and predicts the emotion and gender labels at 97.35 percent. In addition to predicting the exception of random forest and ann mode, this research can also be observed that specific prediction Emotion labels are more accurate than gender and emotion label predictions because only predicted emotion label has only seven unique labels, which is less than the prediction of gender and emotion labels with 14 unique labels. In predicting emotion and gender labels, the female label was more accurate than the male label because the number of female voices at 57.82 percent was slightly more than the number of male voices at 42.18 percent. Therefore, females have more than male voices and the amount of female training slightly more than males. It makes female voice in model prediction slightly more accurate than male that of almost all models. In this research, the surprise sound is the most accurate, while the fear sound is the least accurate. It can be understood that the sound of the surprise is the clearest. At the same time, the sound of fear is quite challenging and has relatively minimal clarity. Although research of T. Seehapoch and S. Wongthanavasu (2013) [45]. and AkÃ§ay, M. B., OÄuz, K. (2020) [2]. with an accuracy of 98 percent is slightly more accurate than this research, the highest being at 97.44 percent. Nevertheless, it is acceptable because This research differs from other research in 4 ways. Firstly, This research uses 17 models divided into 11 machine learning models and six deep learning models to compare and find the best model because the study saw that most research used only 1 to 3 models. Secondly, This research uses four datasets because, in many datasets, there are too many male or female voices, but in this dataset, the number of male and female voices is similar. Neutral emotion is slightly less than other emotions, and only surprise emotion is much less than other datasets. The inclusion of datasets in this research can help solve the problem of overfitting and imbalanced data. Thirdly, Do the research and compare the data we want to predict, just emotion labels and data that intends to predict gender and emotion labels simultaneously, which in most research studies will only compare the emotion. Very few compare both gender and emotion labels. However, this research will compare both at the same time. Fourthly, ensemble learning is used to bring many models to vote, which makes the model more accurate, which is not present in most studies. However, this research has a limitation in that we cannot tune parameters with GridSearchCV in machine learning because it takes too long to complete the run, as can be seen from the figure 7.1 and figure

7.2.which shows one run per parameter used, so a long time and it makes do not have the best parameters to make our machine learning model as accurate as possible. In addition, deep learning modeling takes a very long time. It makes for a limited number of daily trials because running takes a long time. In the future, research run with good GPU will be more accurate as it will be possible to tune parameters with GridSearchCV and it can try out deep learning models much faster for training models many times.

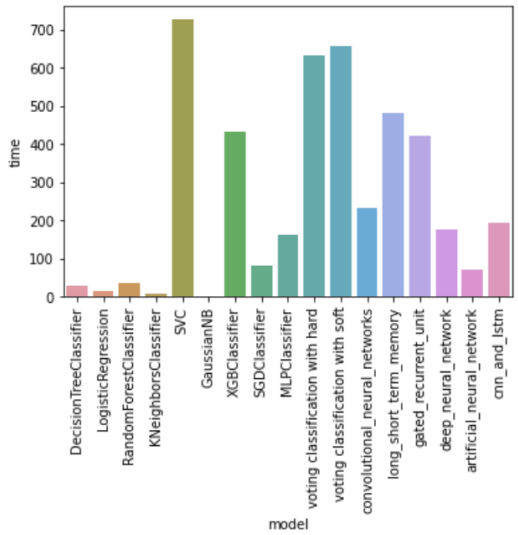

Figure 7.1: train model per minute in every model with just emotion labels

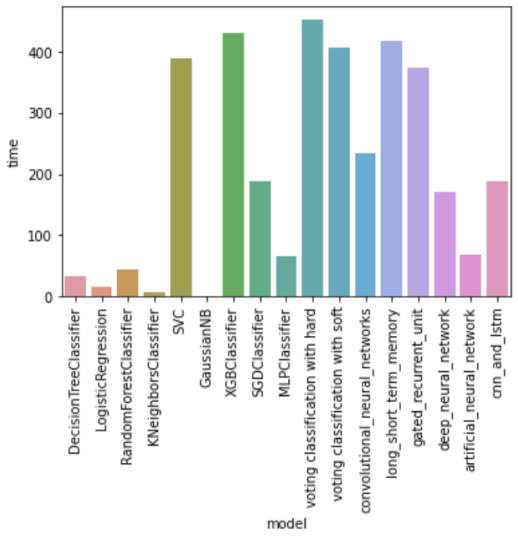

Figure 7.2: train model per minute in every model with gender and emotion labels

CHAPTER 8

Conclusion

This research studies and researches emotional analysis from the sound. Which uses the emotional data set from speech four datasets are Ravdess dataset, Crema dataset, Savee dataset, and TESS dataset. The dataset contains seven emotions: happiness, disgust, fear, happy, neutral, sad, and surprise. this research use three feature extraction Zero Crossing Rate, Root Mean Square, and the Mel Frequency Cepstral Coefficient, which can be extracted from a vector of 2376 columns. After that, we normalize before creating the model. this study train model all 17 models. The least accurate predictive emotion label is fear emotion but the best emotion to predict is a surprise. In addition, the prediction model Female voices are more accurate than male voices. In this study, the most accurate model for a machine learning model was the ensemble learning with the soft vote, with only an emotion labeling accuracy of 89.82 percent and an emotion and gender label prediction accuracy of 88.64 percent. While the most accurate prediction in deep learning is Convolutional neural network predicted only emotion label accuracy was 97.44 percent, and predicted by emotion and gender accuracy was 97.35 percent. We look at the deeper predictions. It can be seen clearly that the accuracy of females will have an accuracy of 98.32 percent, which is more than males with an accuracy of 96.17 percent, which is not the only model where females have more accuracy than males, but all models. This analysis is likely due to the dataset in this research, female voices are more pronounced than male voices, and the number of female voice data is slightly more than that of males.

Bibliography

[1] Gaurav Agarwal and Hari Om. 2021. Performance of deer hunting optimization based deep learning algorithm for speech emotion recognition. *Multimedia Tools and Applications* 80, 7 (2021), 9961–9992.

[2] Mehmet Berkehan Akçay and Kaya Oğuz. 2020. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication* 116 (2020), 56–76.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*. Ieee, 1–6.

[4] Enrique M Albornoz, Diego H Milone, and Hugo L Rufiner. 2011. Spoken emotion recognition using hierarchical classifiers. *Computer Speech & Language* 25, 3 (2011), 556–570.

[5] Gokhan Altan, Sertan Alkan, and Dumitru Baleanu. 2022. A novel fractional operator application for neural networks using proportional Caputo derivative. *Neural Computing and Applications* (10 2022). DOI:http://dx.doi.org/10.1007/s00521-022-07728-x

[6] Hadhami Aouani and Yassine Ben Ayed. 2020. Speech emotion recognition with deep learning. *Procedia Computer Science* 176 (2020), 251–260.

[7] Ismail Babajide Mustapha and Faisal Saeed. 2016. Bioactive molecule prediction using extreme gradient boosting. *Molecules* 21, 8 (2016), 983.

[8] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30071–30078.

[9] Sagar K. Bhakre and Arti Bang. 2016. Emotion recognition on the basis of audio signal using Naive Bayes classifier. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2363–2367. DOI:http://dx.doi.org/10.1109/ICACCI.2016.7732408

[10] Dmitri Bitouk, Ragini Verma, and Ani Nenkova. 2010. Class-level spectral features for emotion recognition. *Speech communication* 52, 7-8 (2010), 613–625.

[11] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[12] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing* 5, 4 (2014), 377–390.

[13] Rahul Dey and Fathi M Salem. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 1597–1600.

[14] Mehmet Bilal Er. 2020. A novel approach for classification of speech emotions based on deep and acoustic features. *IEEE Access* 8 (2020), 221640–221653.

[15] Jingru He and Liyong Ren. 2021. Speech Emotion Recognition using XGBoost and CNN BLSTM with Attention. In *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*. IEEE, 154–159.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Dias Issa, M Fatih Demirci, and Adnan Yazici. 2020. Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control* 59 (2020), 101894.

[18] Agnes Jacob. 2017. Modelling speech emotion recognition using logistic regression and decision trees. *International Journal of Speech Technology* 20, 4 (2017), 897–905.

[19] Dipti D Joshi and MB Zalte. 2013. Speech emotion recognition: a review. *IOSR J. Electron. Commun. Eng.(IOSR-JECE)* 4, 4 (2013), 34–37.

[20] Piotr Juszczak, D Tax, and Robert PW Duin. 2002. Feature scaling in support vector data description. In *Proc. asci*. Citeseer, 95–102.

[21] Leila Kerkeni, Youssef Serrestou, Mohamed Mbarki, Kosai Raoof, and Mohamed Ali Mahjoub. 2018. Speech Emotion Recognition: Methods and Cases Study. *ICAART (2)* 20 (2018).

[22] Anders Krogh. 2008. What are artificial neural networks? *Nature biotechnology* 26, 2 (2008), 195–197.

[23] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, and Julien Epps. 2019. Direct modelling of speech emotion from raw speech. *arXiv preprint arXiv:1904.03833* (2019).

[24] Michael P LaValley. 2008. Logistic regression. *Circulation* 117, 18 (2008), 2395–2399.

[25] Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. 2011. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication* 53, 9-10 (2011), 1162–1171.

[26] Zhiyuan Li and Sanjeev Arora. 2019. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454* (2019).

[27] Zhen-Tao Liu, Qiao Xie, Min Wu, Wei-Hua Cao, Ying Mei, and Jun-Wei Mao. 2018. Speech emotion recognition based on an improved brain emotion learning model. *Neurocomputing* 309 (2018), 145–156.

[28] Qirong Mao, Ming Dong, Zhengwei Huang, and Yongzhao Zhan. 2014. Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE transactions on multimedia* 16, 8 (2014), 2203–2213.

[29] Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications* 5 (2001), 64–67.

[30] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083* (2010).

[31] Michael A Nielsen. 2015. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA.

[32] Fatemeh Noroozi, Tomasz Sapiński, Dorota Kamińska, and Gholamreza Anbarjafari. 2017. Vocal-based emotion recognition using random forests and decision tree. *International Journal of Speech Technology* 20, 2 (2017), 239–246.

[33] Oluwaseun Oyebode and Desmond Ighravwe. 2019. Urban Water Demand Forecasting: A Comparative Evaluation of Conventional and Soft Computing Techniques. *Resources* 8 (09 2019), 156. DOI:http://dx.doi.org/10.3390/resources8030156

[34] Ripon Patgiri, Sajid Hussain, and Aditya Nongmeikapam. 2020. Empirical Study on Airline Delay Analysis and Prediction. (02 2020).

[35] Leif E Peterson. 2009. K-nearest neighbor. *Scholarpedia* 4, 2 (2009), 1883.

[36] Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, 55–69.

[37] Anuja Priyam, GR Abhijeeta, Anju Rathee, and Saurabh Srivastava. 2013. Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology* 3, 2 (2013), 334–337.

[38] Md. Mijanur Rahman, Md Al-Amin, and Md Bhuiyan. 2012. Continuous Bangla Speech Segmentation using Short-term Speech Features Extraction Approaches. *International Journal of Advanced Computer Science and Applications* 3 (11 2012). DOI:http://dx.doi.org/10.14569/IJACSA.2012.031121

[39] TM Rajisha, AP Sunija, and KS Riyas. 2016. Performance analysis of Malayalam language speech emotion recognition system using ANN/SVM. *Procedia Technology* 24 (2016), 1097–1104.

[40] Payam Refaeilzadeh, Lei Tang, and Huan Liu. 2009. Cross-validation. *Encyclopedia of database systems* 5 (2009), 532–538.

[41] Yuan REN and Guangchen BAI. 2011. New Neural Network Response Surface Methods for Reliability Analysis. *Chinese Journal of Aeronautics* 24 (02 2011), 25–31. DOI:http://dx.doi.org/10.1016/S1000-9361(11)60004-6

[42] Jia Rong, Gang Li, and Yi-Ping Phoebe Chen. 2009. Acoustic feature selection for automatic emotion recognition from speech. *Information processing & management* 45, 3 (2009), 315–328.

[43] Sandra Rothe, Sebastian Wirtz, and Dirk SÃ¶ffker. 2016. About the reliability of diagnostic statements: fundamentals about detection rates, false alarms, and technical requirements.

[44] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[45] Thapanee Seehapoch and Sartra Wongthanavasu. 2013. Speech emotion recognition using Support Vector Machines. In *2013 5th International Conference on Knowledge and Smart Technology (KST)*. 86–91. DOI:http://dx.doi.org/10.1109/KST.2013.6512793

[46] James G Shanahan, Yan Qu, and Janyce Wiebe. 2006. *Computing attitude and affect in text: Theory and applications*. Vol. 20. Springer.

[47] Rahul Sharma, Minju Kim, and Akansha Gupta. 2022. Motor imagery classification in brain-machine interface with machine learning algorithms: Classical approach to multi-layer perceptron model. *Biomedical Signal Processing and Control* 71 (2022), 103101.

[48] Jaime Lynn Speiser, Michael E Miller, Janet Tooze, and Edward Ip. 2019. A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications* 134 (2019), 93–101.

[49] Johan AK Suykens. 2001. Nonlinear modelling and support vector machines. In *IMTC 2001. proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, Vol. 1. IEEE, 287–294.

[50] Jimin Tan, Jianan Yang, Sai Wu, Gang Chen, and Jake Zhao. 2021. A critical look at the current train/test split in machine learning. *arXiv preprint arXiv:2106.04525* (2021).

[51] Li Wen and Michael Hughes. 2020. Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques. *Remote Sensing* 12, 10 (2020), 1683.

[52] Alex Witsil and Jeffrey Johnson. 2020. Volcano video data characterized and classified using computer vision and machine learning algorithms. *Geoscience Frontiers* 11 (02 2020). DOI:http://dx.doi.org/10.1016/j.gsf.2020.01.016

[53] Chih-Hung Wu, Yueh-Min Huang, and Jan-Pan Hwang. 2016. Review of affective computing in education/learning: Trends and challenges. *British Journal of Educational Technology* 47, 6 (2016), 1304–1323.

[54] Harry Zhang. 2004. The optimality of naive Bayes. *Aa* 1, 2 (2004), 3.