applied sciences

MDPI

Article

Using Vocal-Based Emotions as a Human Error Prevention System with Convolutional Neural Networks

Areej Alsalhi and Abdulaziz Almehmadi *

Department of IT, Faculty of Computing and IT, AIST Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia
* Correspondence: aalmehmadi@ut.edu.sa

Abstract: Human error is a mark assigned to an event that has negative effects or does not produce a desired result, with emotions playing an important role in how humans think and behave. If we detect feelings early, it may decrease human error. The human voice is one of the most powerful tools that can be used for emotion recognition. This study aims to reduce human error by building a system that detects positive or negative emotions of a user like (happy, sad, fear, and anger) through the analysis of the proposed vocal emotion component using Convolutional Neural Networks. By applying the proposed method to an emotional voice database (RAVDESS) using Librosa for voice processing and PyTorch, with the emotion classification of (happy/angry), the results show a better accuracy (98%) in comparison to the literature with regard to making a decision to deny or allow a user to access sensitive operations or send a warning to the system administrator prior to accessing system resources.

Keywords: CNN; vocal analysis; human error detection

Citation: Alsalhi, A.; Almehmadi, A. Using Vocal-Based Emotions as a Human Error Prevention System with Convolutional Neural Networks. *Appl. Sci.* **2024**, *14*, 5128. https://doi.org/10.3390/app14125128

Academic Editor: Douglas O'Shaughnessy

Received: 4 May 2024
Revised: 8 June 2024
Accepted: 10 June 2024
Published: 12 June 2024

1. Introduction

Although human error is one of the most common causes for accidents in many fields, researchers have become interested in finding the reasons behind it. The results unexpectedly pointed out that one of the key reasons for this issue is emotion. To solve this problem, we explored the suggested solutions to this in the literature, such as testing the employees, limiting their privilege [1,2]. Other solutions that directly apply to the employees may be of use as well; however, one of the common solutions that can be applied on the systems is the non-identity-based biometric model, which uses behavior or physiological data from the human, such as the face, voice, or text, to identify the emotion of the user [3]. This paper proposes to develop a voice-based emotion detection system to detect and prevent human error.

Human error has always been a weak point that impacts the security and accuracy in many systems. For example, the results of human errors resulting from pressure in the work environment can cause emotional or psychological stress, leading people to carry out tasks incorrectly, ultimately being a cause of the loss of lives and property [1,4]. Moreover, it has become an increasingly important factor in security breaches that may affect confidential data, with most cyber data breaches being caused by human errors [5]. According to the literature, human error is the cause of 70% to 90% of accidents in any field [6]. These errors happen accidentally or intentionally, but one of the primary reasons for this is certainly a user's emotions. Markedly, in many fields, emotion can make the user make the wrong decision that would not be made in a normal situation.

This paper aims to create a voice-based system that can recognize a user portraying positive or negative emotions (happy/angry). Additionally, depending on the requested resources, the system will reject/allow the user to access or complete sensitive tasks or send an alert to the system administrator to protect the user from making the wrong decision.

The system builds an emotional profile per user and makes the decision based on the current emotional state of the user. While emotions have a nuanced role in decision making—with both negative and positive emotions potentially having a negative or positive impact on the decision-making process, sometimes leading to better or worse decisions, depending on the context and other various factors—the proposed system serves as the first major step, which is to detect the emotions a user experiences at a specific moment.

The remainder of this paper is organized as follows: In Section 2, we present related research. In Section 3, we present the proposed methodology. In Section 4, we explore the results and finally provide the conclusions.

2. Literature Review

2.1. Human Error

Many human error incidents happen daily, which extends ramifications that can lead to benign results with no effect, versus catastrophic consequences, which can stop the work of a whole business. This enhances the importance of studying human error to reduce their harm to organizations. In ref. [7], the author defines human error as "voluntary and deliberate action by a human interacting with another system that exceeds established tolerances defined by that system". This definition can hold many outlooks which describe a problem and how to solve it.

2.2. Uni-Model Emotion Detection

Emotions are defined in a variety of ways; however, two standards of emotions are prevalent. First, emotions are reactions that humans experience in response to events or situations. The type of emotion that a person experiences is determined by the circumstances that provoke that emotion. For example, a person feels joy when they receive good news and feels afraid when they are threatened; secondly, emotion includes parts of physiology, and affects behavior and perception [8]. As there are many emotions, Paul Ekman [9] identified and outlined the six basic emotions (joy, surprise, sadness, anger, disgust, and fear); however, to make this system more accurate, our system will focus on two emotions (happy/angry) [9]. Uni-Model Emotion Detection can be based on emotions portrayed by the voice, facial emotions, or text emotions.

Voice biometrics are a type of biometric that uses unique properties of the human voice to identify and authenticate how these properties are directly related to the anatomy of the human acoustic system and behavioral characteristics with regard to speech. Many scientists believe that the human voice is a very effective method of biometric authentication and emotional identification, in addition to the ability of systems to recognize it without the need for custom equipment, except for a microphone.

2.3. Voice-Based Emotion Detection

The extraction of voice features and emotions represents a challenging and complex research area, influenced by various factors, such as the individual's physical condition, gender, mental state, and surrounding noise [10], with the extracted features of a female voice also differing from a male voice and the voice of a child. Manasa [11] also explained that it is possible to employ emotional detection systems in various areas of life, such as polygraph systems, contact centers, car systems, robots, and smart applications [11]. As emotional states affect speech features [12], using software similar to the Phonetic and Acoustic Analysis Toolkit (PRAAT) can help in a manual analysis of focal features to detect emotions from speech. Also, it is possible to detect the eight types of human emotions (such as sadness, happiness, anger, etc.) in detail through a system that follows three stages: First, the features are extracted from the database through Librosa [13], which is a Python package that extracts and analyzes important features of simple sound and music. Second, sample training is carried out to match the previously extracted features. Thirdly, the testing and classification of emotional sound samples is unknown [14]. Through the application of a number of automated learning algorithms, including the trees algorithm

and K-nearest neighbors (KNN) algorithm, we can obtain the most efficient algorithm between them. The preference emerged after comparing three algorithms to the additional tree work manic algorithm with a resolution of up to 99% [10]. On another hand, based on the accuracy of the results of a previous study, a low frequency produces more accurate emotional information with a high acceptance rate when using the Jitter method [14,15]. Moreover, the experimental results of using an SVM classifier to classify emotion categories for several different languages showed the following accuracies: 78.57%, 79.3, 81.43%, 82.8%, and 89.23% [16]. In ref. [17], the authors proposed building a CNN model for an emotion distinction task with a 71% accuracy. Their model was evaluated using a dataset for a speech emotion recognition system using speech samples, with characteristics being extracted from these speech samples using the Librosa [13] package. Classification performance is based on the extracted characteristics. Thus, we can determine the emotion of the speech signals [18]. Table 1 provides a summary of previous works on voice-based emotion detection.

Table 1. Summary of previous works on voice-based emotion detection.

Reference	Methodology	Dataset	Accuracy
[19]	Segment-level features and DNN; utterance-level features and ELM	IEMOCAP	54.3
[20]	spectrogram; DBN	IEMOCAP	64.78
[21]	Speech features; SVM; DBN	Chinese Academy of Sciences emotional speech database	94.6 (using DBN)
[22]	Prosodic features, spectrum features; ELM	CASIA Chinese emotion corpus	89.6
[23]	Probabilistic echo-state network	WaSeP	96.69
[24]	Spectrogram; Deep Retinal Convolution Neural Networks (DRCNNs)	IEMOCAP	99.25
[25]	ResNet101	RAVDESS	72.34
[25]	SqueezeNet	RAVDESS	71.76

2.4. Emotion-Based Access Control

It is hard to imagine an organization that does not use access control to keep it internally and externally secured. It is known that the definition of access control is allowing authorized people and refusing otherwise to use a certain part of a system. This system uses the access control methods to detect and classify the emotions of authorized users and prevent them if the system suggests that they have bad intentions. In light of Almehmadi et al.'s (2013) [26] study, it is possible to overcome security problems in access control by classifying the intentions associated with the detection of emotions. Moreover, Zhang and others examined the method of authentication using multiple biometrics as one of the solutions to the problem of mono-authentication [21]. They managed to carry this out by developing a biometric authentication system for smartphones based on users' faces and sounds. With a difference between the biological features of a face and sounds, the problem with results being different was solved by applying the maximum–minimum method to normalize the matching score. As the researchers explained through their study, the process of identifying a user depends on the image of their face. This can be performed via the integration of wavelet transformation (WT) and hidden Markov models (HMMs) for both eyes, nose, and mouth, with each of the factors being analyzed. In a study conducted on 64 people, a 75% recognition rate and 25% error rate were found (Janusz Bobulski., 2012) [27]. Emotion-based access control is a new field, so there is a lack of related knowledge about it.

3. Emotion-Based Human Error Detection System

3.1. Human Error

A built-in system was implemented with a GPU to recognize emotion within a user's voice in order to handle human error using a Convolution Neural Network with automated

feature engineering and better accuracy than all relevant works up to date. This was completed through several stages of the proposed system, as shown in Figure 1.

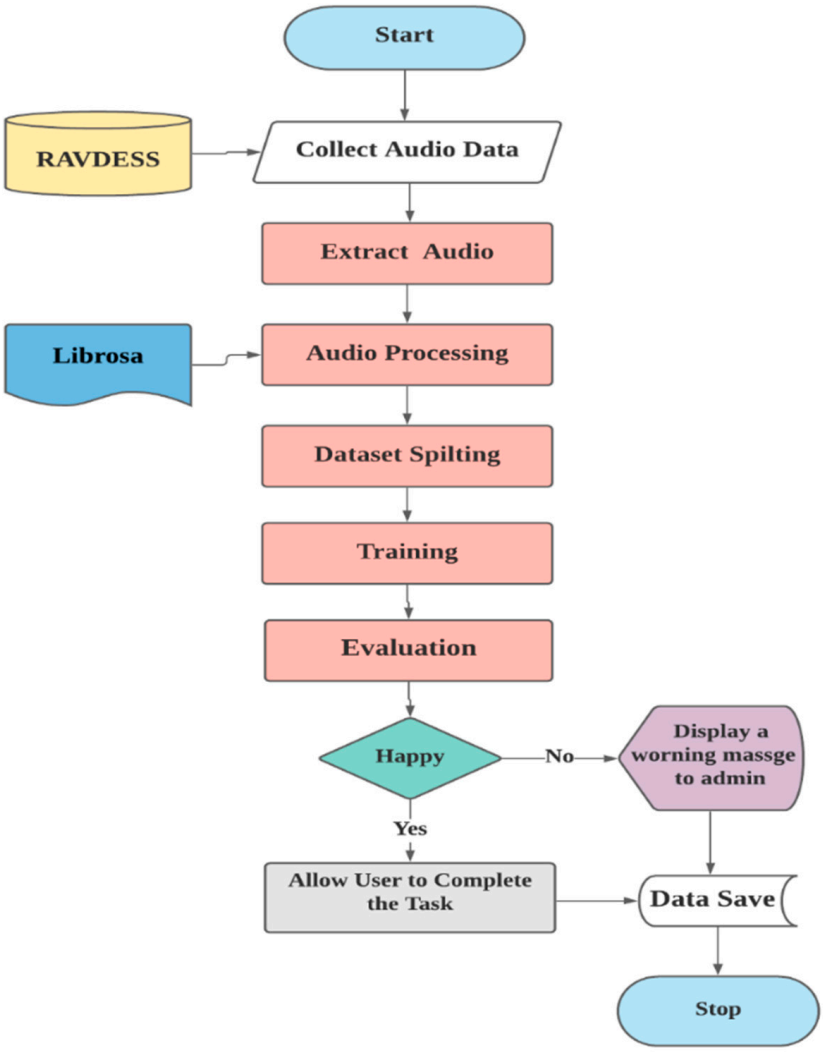

Figure 1. Proposed system.

3.2. Methodology

The system complements an already established system of an organization that has sensitive jobs or activities, such as cybersecurity professionals configuring a firewall, aiming not to impact usability but instead only function when a sensitive decision is being made by the user who has access to sensitive data.

This project identifies the emotions of humans based on their voices. We discuss the proposed model architecture for audio emotion recognition, as shown in Figure 2. For more accurate results and more efficacious findings, it is limited to two emotions: happy and angry. The project is divided into two parts. The first part involves extracting voice features using Librosa [13], which we sampled using the "Kaiser best" sampling algorithm, and we took the first 40 MFCCs for each audio recording, as shown Figures 3 and 4. The second part comprises training the Ryerson Audio-Visual Database using Convolution Neural Networks to classify various audio data as particular emotions.

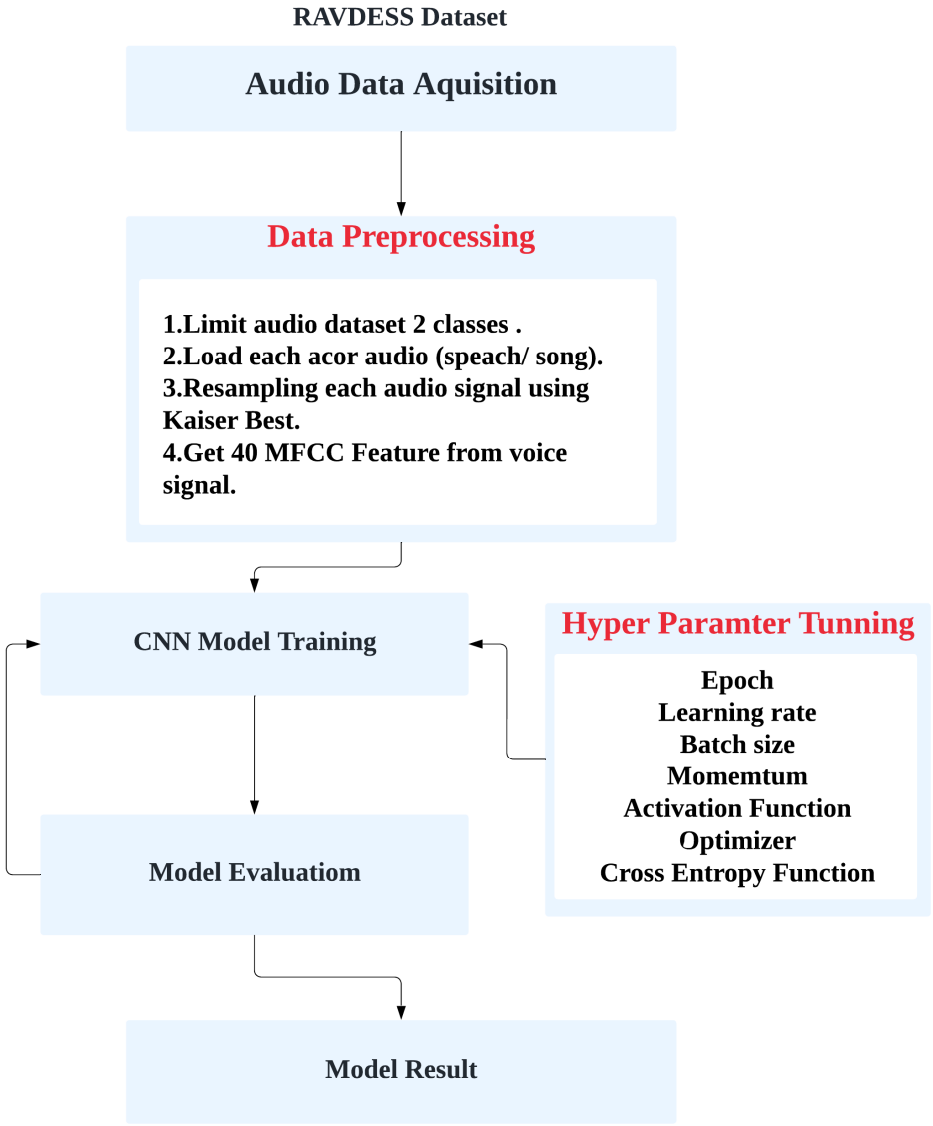

Figure 2. The proposed vocal emotion model architecture.

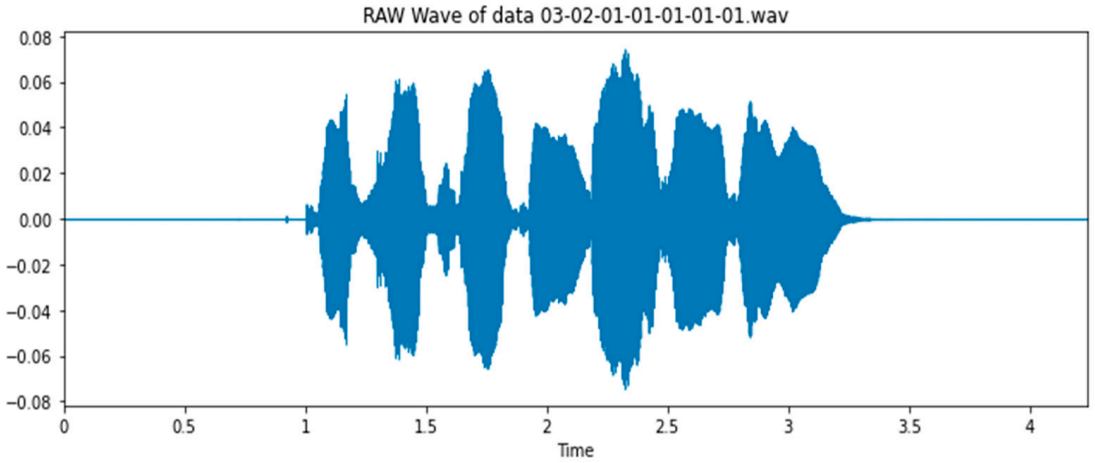

Figure 3. Raw audio signal.

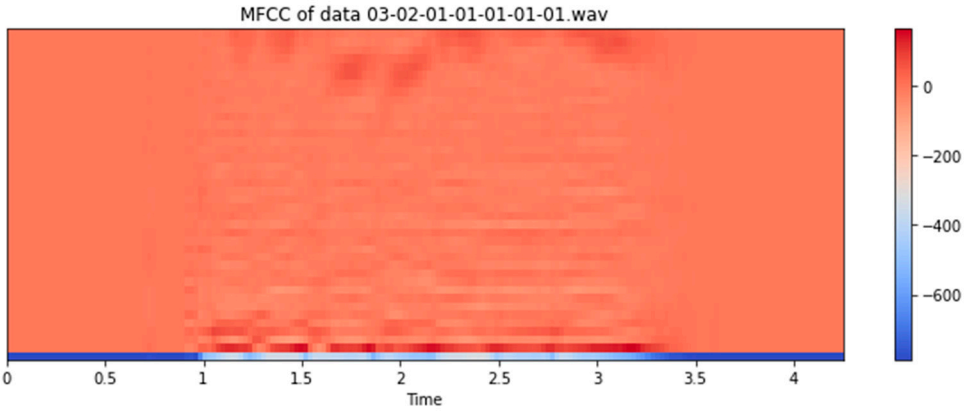

Figure 4. Audio MFCC feature representation.

3.2.1. Dataset

The dataset which will be used for training and testing audio emotion detection is "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)" [28]. This dataset contains a total 24 actors (12 male and 12 female) vocalizing two phonetically matched sentences in a neutral North American accent. It consists of 7356 audio-visual files that are divided into 8 expressions, as shown in Table 1 (audio total size: 1 GB). Table 2 displays the number of samples in the database. The filename is unique, consisting of a 7-part numerical identifier. File parts are structured in the following sequence (modality, vocal channel, emotion, emotional intensity, statement, repetition, and actor) [29].

Table 2. RAVDESS dataset.

Emotions	Speech File 24 Actors 60 Trials per Actor	Song File 23 Actors 44 Trials per Actor
Natural	96	92
Calm	192	184
Happy	192	184
Sad	192	184
Angry	192	184
Fearful	192	184
Surprise	192	0
Disgust	192	0
Total	1440	1012

In the RAVDESS dataset, both speech audio and singing audio files are available. However, in this research, only speech was used as an input due to several reasons and nuances, the most important of which is speech. Speech is primarily used to communicate and transmit information, thoughts, and emotions through spoken language, with the intensity and dynamics of speech naturally varying based on a person's engagement in a conversation and emotional context.

3.2.2. Model

For emotion recognition from audio data, we used an audio analysis library, named Librosa, in Python to extract 40 MFCC features from an audio dataset and save them as an array with the NumPy library. We built a Deep Neural Network model, as shown in Figure 4, using the PyTorch [30] library for classifying the corresponding emotions from an extracted audio Mel-Frequency Cepstral Coefficient (MFCC), which is a popular tool for extracting sound features from sound signals in several steps to simulate the cochlea of

the human ear (Figure 5) [31,32]. Finally, the sound received by the cochlea is calculated as Equation (1):

$$c(n) = \sum_{m=0}^{M-1} log_{10}(\mathrm{s(m)})\cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad (1)$$

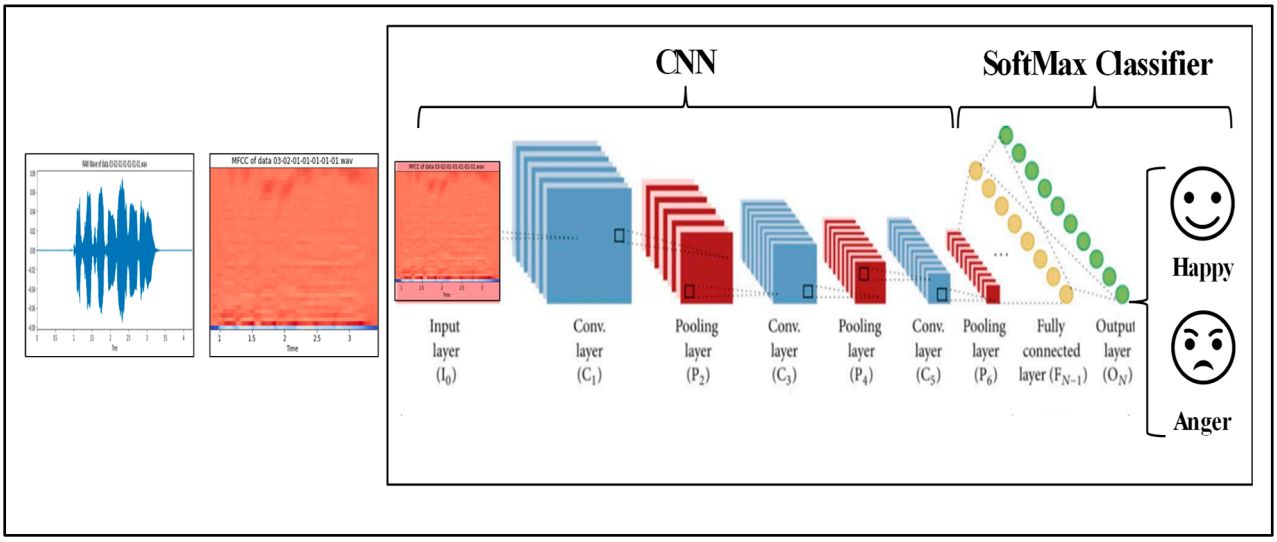

Figure 5. Deep Convolution Neural Network model.

The architecture comprises a Convolutional Neural Network consisting of three convolutional layers with max pooling, aimed at reducing input dimensionality to prevent overfitting. Dropout and batch normalization layers are employed for regularization. Subsequently, the output is flattened and connected to three fully connected layers to predict emotion. During forward propagation, which is the flow of data from input cells towards the output cells, the RELU activation function (a linear function) extracts positive inputs directly. Otherwise, if this does not occur, the result is zero.

4. Experimental Setup and Results

4.1. Experimental Setup

We took the collected data for emotion recognition from audio data based on Section 3.1 and used an 80-10-10 split, with 80% of the data being used for training and 10% randomly used for testing and validation, as shown in Figure 6. We trained our Convolution Neural Network model using a stochastic gradient descent optimizer, 0.9 momentum, 32 batch size, and a cross-entropy loss function, and we evaluated our model with different epochs. This allowed us to measure the number of times the learning algorithm could operate run through the whole training dataset. Moreover, the learning rate (0.01, 0.001) indicates the size of the learning step used in the training of Neural Networks. All experiment were run on the GPU device using Colab [8] using Librosa [13] and PyTorch. Full source code is available at Appendix A.

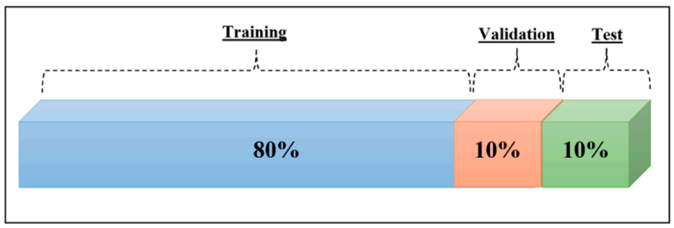

Figure 6. Dataset splitting.

4.2. Results

We checked the accuracy of training, validation, and testing. As shown in Figure 7 and Table 3, the rate of learning controls shows us how easily the model adapted to the problem. Due to the smaller shifts, smaller learning rates require more training time. Modifications were continuously made to the weights, although larger learning rates resulted in rapid improvements and required fewer epochs of training. After tuning the learning rate and epochs with momentum, we observed the highest accuracy for our trained model when we chose a small learning rate = 0.001 with 1000 training epochs.

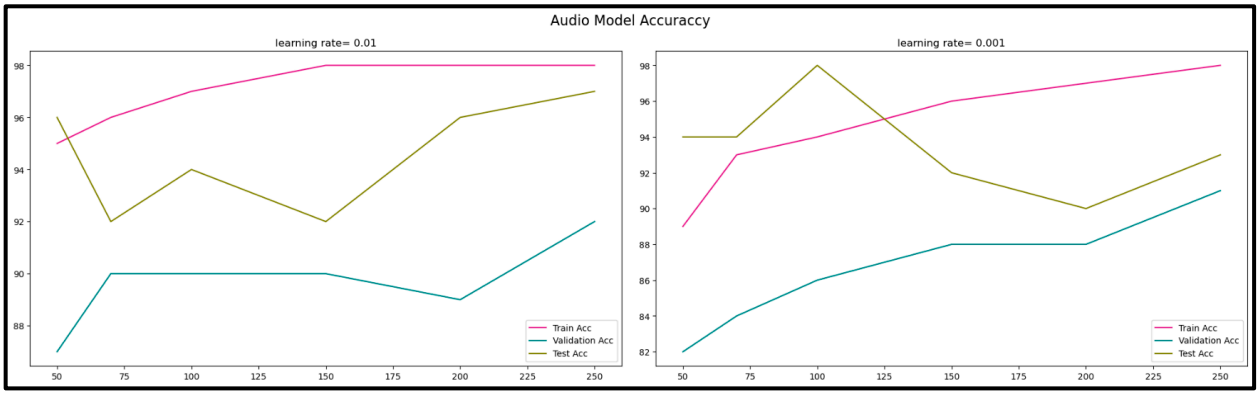

Figure 7. Audio model accuracy.

Table 3. Model results.

Learning Rate	**0.01**						**0.001**					
Epoch	50	70	100	150	200	250	50	70	100	150	200	250
Validation	87	90	90	90	89	92	82	84	86	88	88	91
Training	95	96	97	98	98	98	89	93	94	96	97	98
Testing	96	92	94	92	96	97	94	94	98	92	90	93

5. Conclusions

In this research, we proposed voice emotion recognition for classifying speech using the RAVDESS database. The main target of this technique is to obtain the highest accuracy when classifying a sample using MFCC features captured from speech. The proposed model was used for reducing human error by either granting or denying access control to sensitive data to the user based on their voice. The model presented a high accuracy of 98% according to previous works. It is a very efficient binary classification technique for voice emotion recognition, which can be adapted to many companies to assess their customer reactions through their voices.

The future goal is to improve the performance and efficiency of the system and algorithms in detecting emotions, helping to reduce human errors. This can be achieved by continuing research on the eight types of emotions and by integrating access control systems for voice recognition and facial recognition. Further, applying voice sample cleaning and normalization may improve the current results and reduce false positives. Also, ethical implications may arise since the technology always requires recording audio. Even though the system only analyzes the emotions of tone and not the message conveyed in speech, it still can an ethical dilemma if not addressed properly. Finally, detecting the level of an emotion may play an important role in determining how much it impacts decision making, which may be a valuable factor for detecting and preventing human error as well as being a valuable improvement to the accuracy of the current system.

Author Contributions: Conceptualization, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); methodology, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); software, A.A. (Areej Alsalhi); validation, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); formal analysis, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); investigation, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); resources, A.A. (Abdulaziz Almehmadi); data curation, A.A. (Areej Alsalhi) and A.A. (Abdulaziz Almehmadi); writing—original draft preparation, A.A. (Areej Alsalhi); writing—review and editing, A.A. (Abdulaziz Almehmadi); visualization, A.A. (Areej Alsalhi); supervision, A.A. (Abdulaziz Almehmadi); project administration, A.A. (Abdulaziz Almehmadi); funding acquisition, A.A. (Abdulaziz Almehmadi). All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Artificial Intelligence and Sensing Technologies Research Center at the University of Tabuk, grant number 1445-200.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Human Error Prevention System Based on Vocal Emotions Using Convolutional Neural Network Code

Feature Extract

```
import os
import librosa
import numpy as np
audio_path = 'drive/My Drive/audiomodel/Audiodataset'
tmpdir = 'drive/My Drive/audiomodel/temp'
def preprocess(audio_path, tmpdir):
    audio_store_path = tmpdir + '/' + 'aud_features/'
    make_dir_aud = False

    if not os.path.exists(audio_store_path):
        os.mkdir(audio_store_path)
        make_dir_aud = True

    if (not make_dir_aud):
        return

    for subdir, dirs, files in os.walk(audio_path):
        for file in files:
            if  file[6:8] in ["03","05"] :
                print(file)
                try:
```

```
                    X,                  sample_rate                  =
librosa.load(os.path.join(subdir,file), res_type='kaiser_best')
                    mfccs    =    np.mean(librosa.feature.mfcc(y=X,
sr=sample_rate, n_mfcc=40).T,axis=0)
                    mfccs = np.asarray(mfccs)
                    np.save(audio_store_path + file[0:-4], mfccs)
                except ValueError:
                    continue
```

Data split

```
import os
from sklearn.model_selection import train_test_split
import numpy as np
aud_file_path = 'drive/My Drive/audiomodel/temp/aud_features/'
folders = [ '3',  '5']
def create_test_train_val_sets(tmpdir):

    aud_file_path = tmpdir + '/' + 'aud_features/'

    aud_train_set = tmpdir + '/' + 'aud_train/'

    aud_val_set = tmpdir + '/' + 'aud_val/'

    aud_test_set = tmpdir + '/' + 'aud_test/'

    created = False
    paths = [aud_train_set, aud_val_set, aud_test_set]

    for path in paths:
        if not os.path.exists(path):
            os.mkdir(path)
            created = True
            for i in folders:
                subfolder = path + i
                os.mkdir(subfolder)

    print(created)
    if not created:
        return

    aud_data = []

    aud_files = []
```

```
for file in os.listdir(aud_file_path):
    aud = np.load(os.path.join(aud_file_path, file))
    aud_data.append(aud)
    aud_files.append(file)

aud_train, aud_test, aud_train_files, aud_test_files = train_test_split(aud_data, aud_files, test_size=0.2, random_state=42)
aud_val, aud_test, aud_val_files, aud_test_files = train_test_split(aud_test, aud_test_files, test_size=0.5, random_state=42)


print("starting to write audio data")
print(len(aud_train),"TR")
for i in range(len(aud_train)):
    aud = aud_train[i]
    aud = np.reshape(aud, (1, 40))
    fp = aud_train_set + str(int(aud_train_files[i][7:8]) ) + '/'
    np.save(fp + "aud%d" %i, aud)

print(len(aud_test),"Test")
for i in range(len(aud_test)):
    aud = aud_test[i]
    aud = np.reshape(aud, (1, 40))
    fp = aud_test_set + str(int(aud_test_files[i][7:8])) + '/'
    np.save(fp + "aud%d" %i, aud)


print(len(aud_val),"VAL")

for i in range(len(aud_val)):
    aud = aud_val[i]
    aud = np.reshape(aud, (1, 40))
    fp = aud_val_set + str(int(aud_val_files[i][7:8]) ) + '/'
    np.save(fp + "aud%d" %i, aud)
```

Audio model

```
from __future__ import print_function, division
import torch
import torch.nn as nn
import torch.nn.functional as F
```

```
import torchvision as tv
import torch.optim as optim
import numpy as np
from torchsummary import summary
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.aud_conv1 = nn.Conv1d(1, 16, 3)
        self.aud_pool = nn.MaxPool1d(2)
        self.aud_conv2 = nn.Conv1d(16, 32, 3)
        self.aud_conv3 = nn.Conv1d(32, 64, 3)
        self.fc1 = nn.Linear(64*15, 120)
        self.d1 = nn.Dropout(0.2)
        self.bn1 = nn.BatchNorm1d(64*15)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 2)

    def forward(self, x_aud):
        x_aud = self.aud_pool(F.relu(self.aud_conv1(x_aud)))
        x_aud = F.relu(self.aud_conv2(x_aud))
        x_aud = F.relu(self.aud_conv3(x_aud))
        x_aud = x_aud.view(-1, 64*15)
        x_aud = self.bn1(x_aud)
        x_aud = self.d1(x_aud)
        x = F.relu(self.fc1(x_aud))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
def train(aud_data_loader, criterion, net, device, optimizer):
    t_acc = []
    t_loss = []
    v_acc = []
    v_loss = []
    for epoch in range (100):
        running_loss = 0.0
        l = 0
        total = 0
        correct = 0
        for i, data in enumerate(aud_data_loader['aud_train']):
            # get the inputs
            aud_inputs, aud_labels = data
            aud_inputs = aud_inputs.type(torch.FloatTensor)
```

```
            aud_inputs,   aud_labels   =   aud_inputs.to(device),
aud_labels.to(device)
            # zero the parameter gradients
            optimizer.zero_grad()
            print(aud_inputs.shape)
            outputs = net(aud_inputs)
            loss = criterion(outputs, aud_labels)
            _, predicted = torch.max(outputs.data, 1)
            total += aud_labels.size(0)
            correct += (predicted == aud_labels).sum().item()
            loss.backward()
            optimizer.step()

            # print statistics
            running_loss += loss.item()
            l += loss.item()
            if i % 2 == 1:
                running_loss = 0.0

    t_loss.append(l)
    t_acc.append(100*correct/total)

    total = 0
    correct = 0
    l = 0
    with torch.no_grad():
        for i, data in enumerate(aud_data_loader['aud_val'],
0):
            aud_inputs, aud_labels = data
            aud_inputs = aud_inputs.type(torch.FloatTensor)
            aud_inputs,  aud_labels  =  aud_inputs.to(device),
aud_labels.to(device)
            outputs = net(aud_inputs)
            loss = criterion(outputs, aud_labels)
            l += loss.item()
            _, predicted = torch.max(outputs.data, 1)
            total += aud_labels.size(0)
            correct += (predicted == aud_labels).sum().item()

    v_loss.append(l)
    v_acc.append(100*correct/total)
```

```
    print('Finished Training')
    np.save('ONLY_AUDIO_VAL_LOSS', v_loss)
    np.save('ONLY_AUDIO_VAL_ACC', v_acc)
    np.save('ONLY_AUDIO_TRAIN_LOSS', t_loss)
    np.save('ONLY_AUDIO_TRAIN_ACC', t_acc)
    print('Accuracy of the network on the valdiation Audio: %d %%' %
    (
            np.sum(v_acc) / len(v_acc)))
    print('Accuracy of the network on the training Audio: %d %%' %
    (
          np.sum(t_acc) / len(t_acc)))

def test(aud_data_loader, criterion, net, device, optimizer):
    correct = 0
    total = 0
    nb_classes = 2
    confusion_matrix = torch.zeros(nb_classes, nb_classes)
    with torch.no_grad():
        for i, data in enumerate(aud_data_loader['aud_test'], 0):
                aud_inputs, aud_labels = data
                aud_inputs = aud_inputs.type(torch.FloatTensor)
                aud_inputs, aud_labels = aud_inputs.to(device),
    aud_labels.to(device)
                outputs = net(aud_inputs)
                _, predicted = torch.max(outputs.data, 1)
                total += aud_labels.size(0)
                correct += (predicted == aud_labels).sum().item()
                for t, p in zip(aud_labels.view(-1), pre-
    dicted.view(-1)):
                    confusion_matrix[t.long(), p.long()] += 1

    print('Accuracy of the network on the test Audio: %d %%' % (
            100 * correct / total))
def npy_loader(path):
    sample = torch.from_numpy(np.load(path))
    return sample
folder = 'drive/My Drive/audiomodel/temp'
def only_audio(folder):
    device = torch.device("cuda:0" if torch.cuda.is_available()
    else "cpu")
    print (device)
```

```
    aud_dataset = ['aud_train', 'aud_test', 'aud_val']

    audio_data = {}
    print(folder)
    for x in aud_dataset:
        audio_data[x] = tv.datasets.DatasetFolder(root=folder +
'/' + x, loader=npy_loader, extensions=('.npy'))

    aud_data_loader = {}
    for x in aud_dataset:
        aud_data_loader[x] = torch.utils.data.DataLoader(au-
dio_data[x], batch_size=32,
                shuffle=True, num_workers=0)
    net = Net()
    summary(net.to(device), (1,40))
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
    train(aud_data_loader, criterion, net, device, optimizer)
    test(aud_data_loader, criterion, net, device, optimizer)
```

Real data test

```
X, sample_rate = librosa.load('drive/My Drive/audiomodel/Audioda-
    taset/realtest/OAF_mouse_angry.wav', res_type='kaiser_best')
mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
    n_mfcc=40).T,axis=0)
mfccs = np.asarray(mfccs)
np.save('drive/My Drive/audiomodel/Audiodataset/realtest/MFCCFea-
    turs', mfccs)
sample = torch.from_numpy(np.load('drive/My Drive/audiomodel/Audi-
    odataset/realtest/MFCCFeaturs.npy'))
type(sample , dtype=torch.double)
s = sample.unsqueeze(0)  # Add batch dimension
model = Net()
model.eval()
s= torch.reshape(s, (1,1,40))
s= s.type(torch.FloatTensor)
output = model(s)  # Forward pass
pred = torch.argmax(output, 1)
print(pred)
```

Make File

```
result = pred.numpy()[0]
from datetime import datetime
userID=1
```

```
Type = "Audio"
if result == 0 :
    Emotion = "Happy"
    print("INFO , Grant Access")
else:
    Emotion = "Angry"
    print("WARNING , Deny Access")
Profiles = 'drive/My Drive/audiomodel/History'
if not os.path.exists(Profiles):
    os.mkdir(Profiles)
with open(Profiles+"/"+str(userID)+".txt", 'w') as fp:
    date = datetime.today().strftime('%Y-%m-%d-%H:%M:%S')
    Record = str(userID) + ' , '+ date + " , "+ Type  + " , " +
    Emotion
    fp.write(Record)
```

References

1. Fan, S.; Zhang, J.; Blanco-Davis, E.; Yang, Z.; Wang, J.; Yan, X. Effects of seafarers' emotion on human performance using bridge simulation. *Ocean. Eng.* **2018**, *170*, 111–119. [CrossRef]
2. Manjunath, K.; Anu, V.; Walia, G.; Bradshaw, G. Training Industry Practitioners to Investigate the Human Error Causes of Requirements Faults. In Proceedings of the 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN, USA, 15–18 October 2018. [CrossRef]
3. Mcfadden, K.L.; Towell, E.R.; Stock, G.N. Critical Success Factors for Controlling and Managing Hospital Errors. *Qual. Manag. J.* **2004**, *11*, 61–74. [CrossRef]
4. Kishan, R.; Jayakodi, S.; Asmone, A.S. Causal relationships of human errors in building maintenance: Findings from Sri Lanka. *Build. Res. Inf. Build. Res. Inf.* **2023**, *52*, 431–445. [CrossRef]
5. Alkhaiwani, A.H.; Almalki, G.A. Saudi Human Awareness needs. A survey in how human causes errors and mistakes leads to leak confidential data with proposed solutions in Saudi Arabia. In Proceedings of the 2021 National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 27–28 March 2021. [CrossRef]
6. Jabon, M.; Ahn, G.; Bailenson, J. Automatically Analyzing Facial-Feature Movements to Identify Human Errors. *IEEE Intell. Syst.* **2011**, *26*, 54–63. [CrossRef]
7. Hansen, F.D. Human Error: A Concept Analysis. *J. Air Transp.* **2007**, *11*, 61–77.
8. Colab Website. Available online: https://colab.research.google.com/ (accessed on 1 July 2020).
9. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
10. Hossain, N.; Naznin, M. Sensing Emotion from Voice Jitter. In Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, New York, NY, USA, 4–7 November 2018. [CrossRef]
11. Manasa, C.; Dheeraj, D.; Deepthi, V.S. Statistical Analysis of Voice Based Emotion Recognition using Similarity Measures. In Proceedings of the 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), Bangalore, India,, 19–20 March 2019. [CrossRef]
12. Kumbhakarn, M.; Sathe-Pathak, B. Analysis of Emotional State of a Person and Its Effect on Speech Features Using PRAAT Software. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 763–767. [CrossRef]
13. Mande, A.; Telang, S.; Dani, S.; Shao, Z. Emotion Detection Using Audio Data Samples. *Int. J. Adv. Res. Comput. Sci.* **2019**, *10*. [CrossRef]
14. Librosa Website. Available online: https://librosa.org/ (accessed on 1 July 2020).
15. Prakash, C.; Gaikwad, V.; Singh, R.R.; Prakash, O. Analysis of Emotion Recognition System through Speech Signal Using KNN & GMM Classifier. *IOSR J. Electron. Commun. Eng.* **2015**, *10*, 55–67.
16. Samantaray, A.K.; Mahapatra, K.; Kabi, B.; Routray, A. A novel approach of speech emotion recognition with prosody, quality and derived features using SVM classifier for a class of North-Eastern Languages. In Proceedings of the 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015. [CrossRef]
17. Singh, G. Challenges in Automatic Emotion Recognition Process. *Int. J. Adv. Res. Comput. Sci.* **2018**, *9*, 72–75. [CrossRef]
18. Singh, A.; Srivastava, K.; Murugan, H. Speech Emotion Recognition Using Convolutional Neural Network (CNN). *Int. J. Psychosoc. Rehabil.* **2020**, *24*, 2408–2416. [CrossRef]

19. Han, K.; Yu, D.; Tashev, I. Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine. In Proceedings of the Interspeech, Singapore, 14–18 September 2014; pp. 223–227.
20. Haytham, M.; Fayek, M.L.; Cavedon, L. Evaluating deep learning architectures for Speech Emotion Recognition. *Neural Netw.* **2017**, *92*, 60–68.
21. Zhang, W.; Zhao, D.; Chai, Z.; Yang, L.T.; Liu, X.; Gong, F.; Yang, S. Deep learning and SVMbased emotion recognition from Chinese speech for smart affective services. *Softw. Pract. Exper.* **2017**, *47*, 1127–1138. [CrossRef]
22. Liu, Z.-T.; Wu, M.; Cao, W.-H.; Mao, J.-W.; Xu, J.-P.; Tan, G.-Z. Speech emotion recognition based on feature selection and extreme learning machine decision tree. *Neurocomputing* **2018**, *273*, 271–280. [CrossRef]
23. Trentin, E.; Scherer, S.; Schwenker, F. Emotion recognition from speech signals via a probabilistic echo-state network. *Pattern Recognit. Lett.* **2015**, *66*, 4–12. [CrossRef]
24. Niu, Y.; Zou, D.; Niu, Y.; He, Z.; Tan, H. A breakthrough in Speech emotion recognition using Deep Retinal Convolution Neural Networks. *arXiv* **2017**, arXiv:1707.09917.
25. Er, M.B. A Novel Approach for Classification of Speech Emotions Based on Deep and Acoustic Features. *IEEE Access* **2020**, *8*, 221640–221653. [CrossRef]
26. Almehmadi, A.; El-Khatib, K. Authorized! Access Denied, Unauthorized! Access Granted. In Proceedings of the SIN '13: The 6th International Conference on Security of Information and Networks, New York, NY, USA, 26–28 November 2013; pp. 363–367. [CrossRef]
27. Bobulski, J. *Access Control System Using Face Image*; Systems Research Institute of the Polish Academy: Warsaw, Poland, 2012; Volume 73, pp. 42–200.
28. Ravdess, Website. Available online: https://smartlaboratory.org/ravdess/ (accessed on 5 July 2020).
29. Livingstone, S.R.; Russo, F.A. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS ONE* **2018**, *13*, e0196391. [CrossRef] [PubMed]
30. Pytorch, Website. Available online: https://pytorch.org/ (accessed on 1 July 2020).
31. Lalitha, S.; Geyasruti, D.; Narayanan, R.; Shravani, M. Emotion Detection Using MFCC and Cepstrum Features. *Procedia Comput. Sci.* **2015**, *70*, 29–35. [CrossRef]
32. Aida, R.; Ardil, C.; Rustamov, S.S. Investigation of Combined use of MFCC and LPC Features in Speech Recognition Systems. *World Acad. Sci. Eng. Technol.* **2006**, *19*, 74–80.