applied sciences

MDPI

Article

DeepDetection: Privacy-Enhanced Deep Voice Detection and User Authentication for Preventing Voice Phishing

Yeajun Kang , Wonwoong Kim, Sejin Lim, Hyunji Kim and Hwajeong Seo *

Division of IT Convergence Engineering, Hansung University, Seoul 02876, Korea
* Correspondence: hwajeong@hansung.ac.kr; Tel.: +82-760-8033

Abstract: The deep voice detection technology currently being researched causes personal information leakage because the input voice data are stored in the detection server. To overcome this problem, in this paper, we propose a novel system (i.e., DeepDetection) that can detect deep voices and authenticate users without exposing voice data to the server. Voice phishing prevention is achieved in two-way approaches by performing primary verification through deep voice detection and secondary verification of whether the sender is the correct sender through user authentication. Since voice preprocessing is performed on the user local device, voice data are not stored on the detection server. Thus, we can overcome the security vulnerabilities of the existing detection research. We used ASVspoof 2019 and achieved an F1-score of 100% in deep voice detection and an F1 score of 99.05% in user authentication. Additionally, the average EER for user authentication achieved was 0.15. Therefore, this work can be effectively used to prevent deep voice-based phishing.

Keywords: voice phishing; deep voice detection; user authentication; privacy preservation; autoencoder; convolutional neural networks

Citation: Kang, Y.; Kim, W.; Lim, S.; Kim, H.; Seo, H. DeepDetection: Privacy-Enhanced Deep Voice Detection and User Authentication for Preventing Voice Phishing. *Appl. Sci.* **2022**, *12*, 11109. https://doi.org/10.3390/app122111109

Academic Editors: Andrea Prati, Konstantinos Rantos, Konstantinos Demertzis and George Drosatos

Received: 6 September 2022
Accepted: 31 October 2022
Published: 2 November 2022

1. Introduction

As the deep learning paradigm has developed, various fields have grown rapidly. However, cases of using it for crimes (e.g., voice phishing) are increasing as well. For example, voice phishing has occurred with synthetic voices that are difficult for humans to distinguish because of the generalized deepfake technology (https://www.vice.com/en/article/pkyqvb/deepfake-audio-impersonating-ceo-fraud-attempt, accessed on 5 September 2022).

As the threat of impersonation attacks using deep voices rises, deep voice detectors are needed to prevent voice phishing.

Deep voice is a deepfake technology that creates synthetic voices. The voice of a specific person can be collected through a social network service (SNS), making it easy to perform voice phishing through deep voices. In particular, voice phishing with deep voices can cause serious damage in that synthetic voices that are difficult to distinguish from real voices are used. Therefore, the importance of deep voice detection has emerged, and many related studies are being conducted. However, these techniques have a security vulnerability in that voice data are stored in the detection server.

In this paper, we propose a deep voice detection system (i.e., DeepDetection) with privacy-enhanced features. The encoder can be distributed to the user's mobile device. When voice data are input, the feature is extracted in the form of `numpy` through the encoder and transmitted to the server. Then, the `numpy` data are input into the detection model to detect deep voices and perform user classification simultaneously.

By authenticating users through the user classification method, voice phishing damage caused by simple impersonation can be prevented. An example of simple impersonation is pretending to be the owner of the number using one of the numbers in the victim's address book. In this scenario, voice phishing can be accomplished easily because there is no doubt that it is someone else (i.e., voice phishing criminal), unless the voice is completely different.

This approach prevents voice phishing in two ways: by performing primary verification through deep voice detection and secondary verification of whether the sender is the correct sender through user authentication. Since the voice preprocessing for the classifier is performed on the user's local device, the voice data are not stored in the verification server, thereby overcoming the security vulnerabilities of the existing technique.

1.1. Contribution

1.1.1. Preprocessing Method to Protect Privacy Using an Autoencoder

For deep voice detection, preprocessing must be performed before classification. However, there is a vulnerability in the way in which voice data are exposed to the server. We designed the encoder of the autoencoder to perform the preprocessing independently by deploying it to the users. The model can be separated, and the preprocessing and classification tasks can be performed on the user's device and the server, respectively. This approach has the following advantages. First, it can overcome the problem of original data exposure that is the vulnerability of existing deep learning-based deep voice detection methods. Since meaningful features are extracted through the autoencoder, effective data can be used for user authentication and deep voice detection.

1.1.2. Method for Performing User Authentication and Deep Voice Detection

We propose the DeepDetection system, which can perform user authentication and deep voice detection using the ASVspoof2019 dataset simultaneously. The fake data of this dataset is generated by speech synthesis technology, similar to the users belonging to the real dataset. We adopted it in our work. To the best of our knowledge, this is the first system that can perform both user authentication and deep voice detection with ASVspoof2019 simultaneously.

Voice phishing damage caused by simple impersonation can be prevented by authenticating users through user classification; that is, voice phishing can be prevented more effectively through user authentication. This method prevents voice phishing in two ways: by performing primary verification through deep voice detection and secondary verification of whether the sender is the correct sender through user authentication. In deep voice detection, an *F*1 *score* of 100% was achieved, and in user authentication, an *F*1 *score* of 99.05% was achieved with the proposed method.

The remainder of this paper is organized as follows. Section 2 presents related works concerning AI, neural networks, deepfakes, and datasets, while Section 3 presents the proposed system (i.e., DeepDetection), with its evaluation given in Section 4. Section 5 concludes the paper.

2. Related Works

2.1. Artificial Neural Networks

2.1.1. Convolutional Neural Networks

The convolutional neural network is a type of technology that combines a convolution operation with an existing neural network. It is designed to imitate human visual processing and shows strong performance in image processing. A CNN consists of convolution layers that extract and reinforce the features from the input data using a kernel (i.e., filter) and a classifier that classifies images based on the extracted features. In the convolution layer, convolution plays a role in extracting an image's features using filters. At this time, the same filter traverses the input data and performs convolution operations. Since a CNN learns the same weights using the same filters, it has the advantage that it takes less time to learn because the number of parameters to learn is relatively small. After performing the convolution operation, it enables faster and more effective learning through the activation function. Pooling simplifies the output by performing nonlinear downsampling and reducing the number of parameters the network learns [1]. Through the convolution layer, the general features are initially extracted, specific and unique features are extracted as the higher-level layers proceed, and the output data are used for the next layer; that is,

particular features are identified while repeating the process of extracting features from the input data with learning based on them. After this, the resulting value in the form of an image is transformed into a one-dimensional array, which is a form that can be classified and is input to the fully connected layer to perform classification. The more diverse the data used for learning are, the more accurate classification becomes possible [2].

There is the one-dimensional CNN (CNN 1D), which is effective for training time series data. In the CNN 2D, which is well known as a neural network for image learning, the kernel moves horizontally and vertically to learn the local features of the input image, whereas in the CNN 1D, the kernel moves in one direction. Due to these characteristics, it is used for learning time series data.

2.1.2. Autoencoder

Figure 1 shows the structure of the autoencoder model. The autoencoder consists of an encoder and a decoder, and it is an artificial neural network trained with an unsupervised learning method that learns without data labels [3]. It first learns the representation encoded in the data and then aims to produce an output value from the learned encoding representation as close as possible to the input data. Thus, the output of the autoencoder is a prediction of the input. The autoencoder generates a latent variable that is the result of extracting the features of the input data through the encoder neural network. After this, the latent variable (i.e., the output of the encoder) is input into the decoder neural network. The decoder reconstructs the original data (i.e., the input data of the encoder) based on the latent variable. Unlike the classifier, it can extract the features of the data from the input and then reconstruct them again. Due to these characteristics, it is mainly used for noise removal [4], data visualization and reconstruction, and semantic extraction.

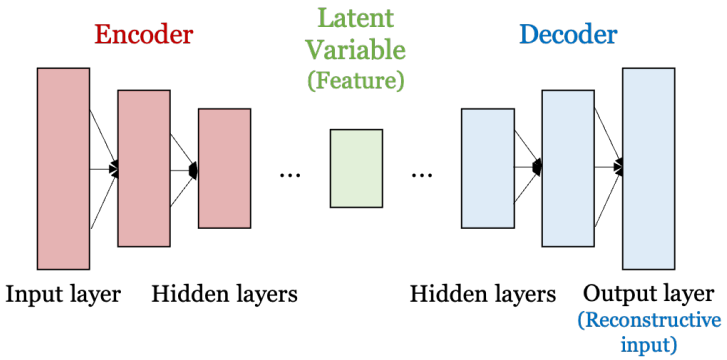

Figure 1. Structure of the autoencoder.

2.1.3. Mean Squared Error

The mean squared error (MSE) is one of the loss functions, and it is used to calculate the difference between the actual value and the predicted value. It is calculated by Equation (1), where n is the total amount of data, y_t is the actual value, and $\hat{y}_t$ is the predicted value. Since the loss function is used to accurately predict the result of the model, it is important to select an appropriate loss function. The MSE is most commonly used in scenarios such as regression problems:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \quad (1)$$

2.1.4. Precision

The *precision* refers to the amount of data that the model predicts to be true and is actually true. It is calculated by Equation (2):

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

2.1.5. Recall

The *recall* is the amount of data that the model recognizes as true for data that are actually true. It is calculated by Equation (3):

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

2.1.6. F1-Score

The *F*1 *score*, also called the f-measure, is used to measure the accuracy in statistical analysis. The existing accuracy equation can achieve high performance in the case of a data imbalance, even when a low-performance model is used. Therefore, the *F*1 *score* is used for accurate calculation results even in the case of a data imbalance. It is calculated as the harmonic mean of the *precision* and *recall*, and it is calculated by Equation (4):

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

2.2. Deepfake

2.2.1. Deepfake

Deepfake is a compound word for deep learning and fake, and it refers to the results of false video, images, and voices generated through the steps of extraction, learning, and generation using deep learning technology. As artificial intelligence advances, sophisticated generation technologies are emerging. Deepfakes can be applied in various fields, but it can also be abused with negative intentions.

2.2.2. Deep Voice

Deep voice is a speech synthesis technology that uses deep learning to create a manipulated voice. As mentioned Section 2.2.1, deepfake is a generic term for non-real, manipulated results generated using deep learning technology. Since it is also widely used in the meaning of video synthesis, in this paper, we call it deep voices instead of deepfakes to focus on speech synthesis technology. Deep voice technology is being applied in real life, such as making audiobooks using the voices of celebrities. Problems that can arise from the misuse of deep voices include legal evidence manipulation and voice phishing. This can cause serious social problems. Research for detecting deep voices is being actively conducted. In Section 2.3, the previous studies on deep voice detection are discussed.

2.3. Dataset

ASVspoof

Automatic speaker verification (ASV) is the authentication of individuals by performing analysis of speech utterances [5,6]. The ASVspoof challenge is a series of challenges that promote the development of countermeasures to protect ASV from the threat of spoofing [7]. The ASVspoof 2019 challenge provides a standard database [8] for anti-spoofing. For our experiment, the logical access (LA) part of ASVspoof 2019 was used. The LA of the provided dataset consists of synthetic speech generated with the very latest, state-of-the-art text-to-speech synthesis (TTS) and voice conversion (VC) technologies, including Tacotron2 [9] and WaveNet [10] as well as bona fide user speech. The spoofing data in the ASVspoof 2019 database have been proven to be similar to the target speaker [8]. Most datasets for deep voice detection consist of deep voices for a single person. However, our proposal is to enhance the prevention of voice phishing by verifying the correct sender after performing verification primarily through deep voice detection. To accomplish this, we need a deep voice dataset for each user. This dataset is suitable for our work as it contains a deep voice for each user. Therefore, we adopted this dataset in order to show the utility of DeepDetection.

2.4. Previous Deep Voice Detection Techniques

In [11], the authors proposed a forensic technique that can distinguish human voices from AI-synthesized voices. The technique is the first bispectral analysis method to distinguish between real and fake voices based on observation of the bispectral artifacts in fake voices.

In [12], the authors introduced a deep voice detection technique based on the theory of detecting subtle differences between real and fake voices. In the process of inputting voice data into the model and processing it as a refined signal, the behavior of neurons is monitored to detect fake voices. A thin ResNet model is used for speaker recognition (SR), and a convolutional layer and a fully connected layer are used to capture the behavior of neurons to classify real speech and fake speech. The captured neurons are input into the binary classifier, and the classifier detects the fake voices.

In [13], the authors used a CNN model to detect fake voices generated using deep voice and imitation techniques, and they used image augmentation and dropout techniques to prevent overfitting. The detection sequence is as follows. When the voice data are input, a histogram is extracted, and image augmentation is performed. The generated dataset is input into the CNN model to perform binary classification that classifies real voices and fake voices. Although there are various methods such as histograms and spectrograms for extracting voice features, the histogram has the advantage that the detection model can classify the voice data without relying on the text. They are characterized by including imitation voices, which are more natural than deep voices and difficult for humans to distinguish, in the fake data.

In [14], through explainable artificial intelligence (XAI), the possibility of interpretation of deepfake detection results is given. Therefore, the authors improved the reliability of the detection results by providing the basis for the detection results through XAI.

2.4.1. Previous Works Using the ASVspoof2019 Dataset

Previous works have focused on detecting real or fake voices and classifying spoofing algorithms generating fake voices. In other words, these are works that verify whether one user is real or fake.

In [15], the authors adapted a light convolutional gated RNN model to improve the long-term dependency for spoofing attack detection.

In [16], the authors used the deep residual network (ResNet [17]) model tho alleviate the gradient loss problem for deep voice detection. Audio preprocessing was performed through a 60-dimension linear filter bank (LFB). To improve generalization performance in deep voice detection, they used a dataset that considered the actual situation and added a FreqAugment layer that randomly masked adjacent data frequencies. Through this work, the model could learn features well even if there were outliers or noise in the dataset.

In [18], the authors proposed a method referred to as feature genuinization, which learns a transformer with a CNN using the characteristics of only a real voice. They then used this genuinization transformer with a light CNN classifier. Their system outperformed other single systems for the detection of synthetic attacks.

In [19], the authors proposed a continuous learning technique called Defining Fake Without Forgetting (DFWF) to detect new spoofing attacks incrementally. Through continuous learning, it was possible to focus more on the characteristics of the real voice while remembering the previously learned information. The problem of forgetting information about the existing data when integrating additional data was addressed with two limitations: Learning without Forgetting (LwF) and Positive Sample Alignment (PSA). They showed that DFWF performed much better than fine-tuning, and the model generated 80-dimension high-level embeddings using a light convolution neural network (LCNN).

In [20], the authors used two types of new acoustic features to improve deep voice detection performance. The first feature consisted of two cepstral coefficients and one LogSpec feature. Cepstral coefficients and LogSpec are extracted from the linear prediction residual signals. The second characteristic is the harmonic and noise subband ratio function.

2.4.2. Security Vulnerabilities in Existing Works

When detecting deep voices with the above-described technique, the original voice data are used as the input for the detection system. At this time, preprocessing is performed on the same server as the detection model. This causes a problem in that the voice data, which are biometric information, are stored in the detection server. Therefore, in this paper, we overcome these security vulnerabilities by separating the server for preprocessing and detection and prevent the original data from being stored on the server.

2.5. Equal Error Rate

The equal error rate (EER) is one of the popular criteria for optimizing speaker verification systems and is a common evaluation criterion for balancing the false rejection rate (FRR) and the false acceptance rate (FAR) [21].

2.5.1. False Rejection Rate

The FRR indicates the false rejection of authorized users. It is calculated by Equation (5). False negative (FN) represents the number of true entries classified as false. True negative (TN) represents the number of false entries classified as false. For example, a user registered in the fingerprint recognition system is not recognized due to an error in the system and is instead recognized as an unauthorized user. Therefore, it is necessary to design a system that minimizes the FRR:

$$FRR = \frac{FN}{FN+TN} \tag{5}$$

2.5.2. False Acceptance Rate

The FAR is an evaluation metric for cases in which unauthorized users are accepted in a specific system, and it is calculated by Equation (6). False positive (FP) represents the number of false entries classified as true. True positive (TP) represents the number of true entries classified as true. For example, the fingerprint of a user who is not registered in the system is mistakenly recognized as another user and accepted:

$$FAR = \frac{FP}{FP+TP} \tag{6}$$

2.5.3. Equal Error Rate

The EER is the value at which the FRR and FAR become equal, and it is a metric that evaluates the overall performance of the biometrics. It is calculated by Equation (7). Since the EER is used as a threshold value of the biometrics, it should be set appropriately according to the sensitivity of the system. As the sensitivity of the system increases, the FRR increases, and the FAR decreases. When the sensitivity is lowered, the FRR decreases, and the FAR increases. It should be set in consideration of the trade-off between the FAR and FRR as well as the importance of the two values in the system:

$$EER = FAR \, or \, FRR, \qquad if \, FAR = FRR \tag{7}$$

3. Proposed Method

The proposed system extracts features through an autoencoder from the voice data in a user's mobile device. After that, the extracted features are transmitted to the server and input into the classifier, and deep voice detection and user authentication are performed. Figure 2 is a schematic diagram of the proposed method, which is shown in a comprehensive manner. As for the specific process, after receiving the voice data as the input, the size of the voice signal is unified through data preprocessing. The preprocessed data are input into the autoencoder. During the encoding, meaningful features are extracted. The encoder can be deployed on users' devices. As shown in Figure 2, when the user's mobile and server environments are divided, only the preprocessed data and not the original voice are stored in the server. Since the original voice is not exposed, privacy can be protected.

The extracted feature vector is input into the classifier, and the classifier classifies the input voice data as users or deep voices. Through this process, user authentication and deep voice detection are possible simultaneously.

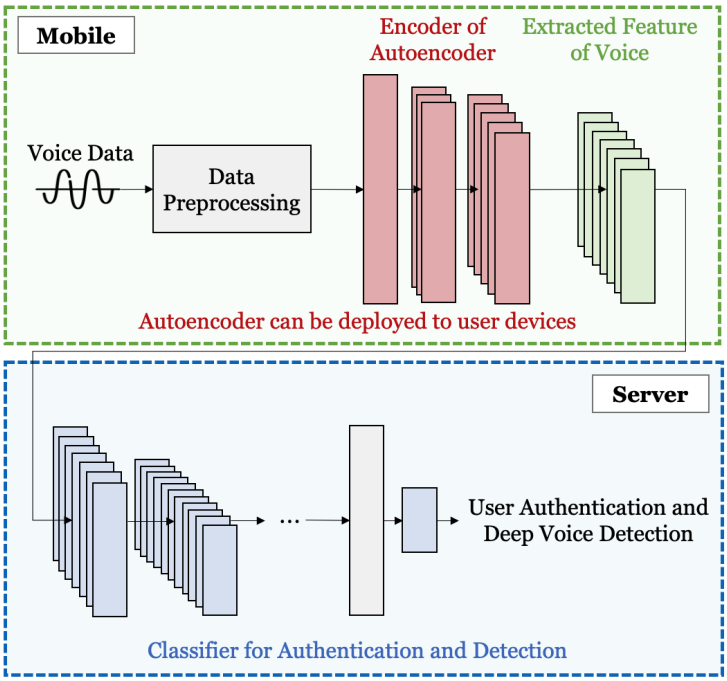

Figure 2. Diagram of proposed method.

3.1. Dataset

3.1.1. Dataset Configuration

Table 1 shows the details of the dataset for the proposed system.

We classified 20 users and a deep voice. In other words, these entries were divided into a total of 21 classes. In order to construct a dataset suitable for our system, the deep voice data were sampled at the same ratio from each user's deep voice. Then, the deep voice data were composed into the same class, regardless of whose deep voice it was. The dataset composed of these criteria was preprocessed for training and inference.

Table 1. Details of dataset (U, D : the number of users and deep voices).

Category	Original Data	Autoencoder	Classifier
Number of classes	40 (20 (U), 20 (D))	21 (20 (U), 1 (D))	21 (20 (U), 1 (D))
Amount of data	25,373	130,516	130,516
Length of sequence	Different for each data	1000	163

3.1.2. Preprocessing

The voices were the time series data, and the raw audio could not be used as the input for the neural network. Therefore, we used preprocessing to convert the raw audio into a `numpy` array for use as input for the neural network. Because the sequence length of the original voice data exceeded at least 50,000, a very large amount of RAM was required for training and inference, and an out of memory (OOM) state occurred in our system with 50 GB of RAM. Since the length of each voice datum was different, it was necessary to unify the sequence length. In order to reduce the length of the sequence of voice data, as long as there was no performance degradation, the length was sliced to 1000. Through this, we reduced the overhead and increased the amount of voice data from 2450 to 130,516. A total of 130,516 data were divided into 60% for training, 20% for validation, and 20% for testing.

3.2. Autoencoder for Feature Extraction from the Voice Data

Architecture of the Autoencoder

Figure 3 and Table 2 show the structure of the autoencoder for feature extraction and details on the hyperparameters of the autoencoder, respectively. In this paper, an autoencoder is used to extract features from the voice data. In DeepDetection, this autoencoder is used for preprocessing of user authentication and deep voice detection. The structure of the autoencoder consists of a Conv1d layer, MaxPool1d layer, and MaxUnpool1d layer. The Conv1d layer is a convolutional layer for sequential data, extracting the features of the input data and making a feature map from them. MaxPool1d is a maxpooling layer for the sequential data which reduces the dimensions of the feature map and prevents overfitting. MaxUnpool1d is opposite of MaxPool1d, and it is used to restore the dimensions reduced through MaxPool1d's encoder in the decoder. The encoder consists of two Conv1d layers and one MaxPool1d layer. The input, hidden, and output channels of the Conv1d layers are 1, 16, and 8, respectively. The length of the data is reduced to 1000, 988, and 978 through the Conv1d layers. The output values of Conv1d go through the ReLU function, which is an activation function that adds nonlinearity, and then are used as the input values of the MaxPool1d layer. The kernel size and stride of MaxPool1d are 6, and the output value has a data length of 163. The decoder consists of two Conv1d layers and one MaxUnpool1d layer. The input is the same as the output of the encoder, and the shape of the decoder is identical to the encoder. Training was performed in 100 epochs, and we achieved an average loss of 0.00076 in classification for 20 users and the fake voice.

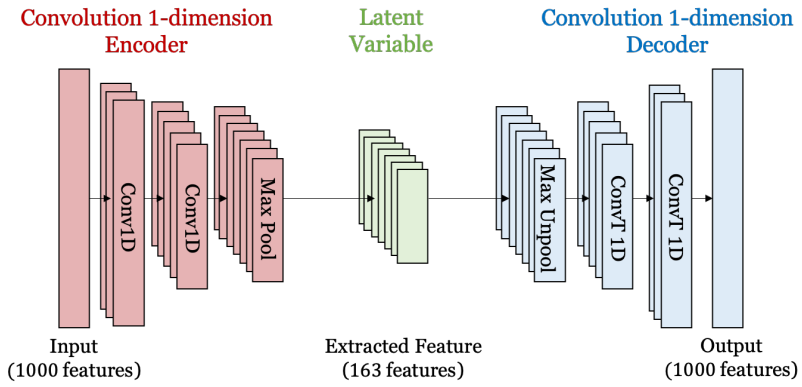

Figure 3. Structure of autoencoder for feature extraction.

Table 2. Details of hyperparameters of autoencoder.

Hyperparameters	Encoder	Decoder
Neurons of input/output layers	1000/163	163/1000
Channels of hidden layers	16	
Channels of latent variable	8 (output of encoder and input of decoder)	
Sequence length	1000	
Kernel size	13, 11	
Strides	1	
Batch size	8	
Loss function	Mean square error	
Activation function	ReLu	
Optimizer	Adam (lr = 0.0001)	
Epoch	100	

3.3. Convolutional Neural Network for User Authentication and Deep Voice Detection

Figure 4 is a diagram showing the structure of a neural network for user authentication and deep voice detection. A vector extracted from speech through a separate autoencoder is input into the classifier. In the scenario, the autoencoder is deployed on a mobile device. The classifier classifies a total of 21 classes including a deep voice and `User 1`∼`User 20`. The vector input into the classifier goes through four one-dimensional CNNs. After that, a flattening operation is performed to be input into the fully connected layer. Unlike a two-dimensional CNN, a one-dimensional CNN is a neural network mainly used for processing sequential data. Our dataset was sequential data because it was voices. Therefore, we used a one-dimensional CNN. The input vector goes through four one-dimensional CNNs. After that, a flattening operation is performed to input the vector into the fully connected layer. After flattening the vector, it is input into two fully connected layers. The output layer does not include the softmax activation function for multiple classifications. Since the softmax activation function is applied to the cross-entropy loss function provided by PyTorch, a separate activation function is not used for the output layer. Table 3 shows the details of the hyperparameters. These are the results derived through hyperparameter tuning. The input shape of the classifier was (8, 163), and the number of neurons in the output layer was 21. Since the number of labels was 21, the number of neurons in the output layer was set to 21. The output channels of each Conv1D layer were 32, 64, 32, and 8. The number of output neurons in the fully connected layer was 64 and 21. The filter size of the convolutional layer was 15, and the stride was 1. After that, it was flattened and made into a 1-dimensional array of a length of 856. Since it was a multi-class classification, cross-entropy loss was used, and Adam with a learning rate of 0.0001 was used as the optimization function. Finally, the epoch was set to 50, and the batch size was set to 128.

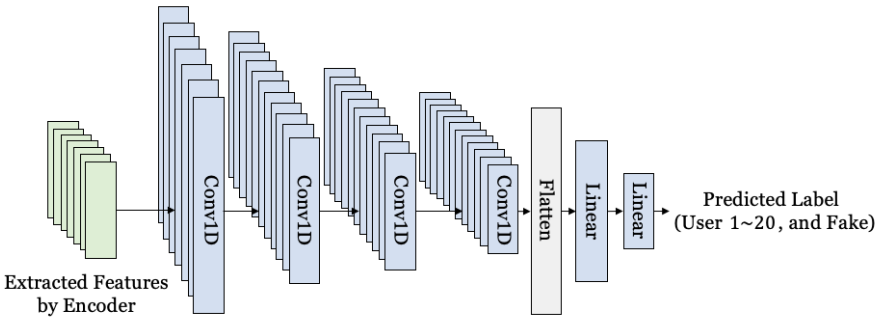

Figure 4. Structure of classifier for authentication and detection.

Table 3. Details of hyperparameters of CNN.

Hyperparameters	Descriptions
Shape of Input and Output	Input (8, 163), Output (21)
The Number of Labels	21 (1 Deep Voice and 20 Users)
Neurons of Layers	Conv1D (149, 135, 121, 107), Flatten (856), Linear (64, 21)
Channel of Layers	Conv1D (32, 64, 32, 8)
Kernel Size	15
Strides	1
Batch Size	128
Loss Function	Cross-Entropy Loss
Activation Function	ReLU
Optimizer	Adam (lr = 0.0001)
Epoch	50

4. Evaluation

4.1. Experiment Environment

We used Google Colaboratory Pro+, a cloud-based service, for this experiment. The operating system was Ubuntu 18.04.5 LTS, and the RAM was 51 GB. The GPU was a Tesla P100 16 GB, and the version of CUDA was 11.1. We used Python 3.7.13 and PyTorch 1.11.0+cu113.

4.2. Reconstruction the Rate of Voice Data Using an Autoencoder

Figure 5 shows the original data and the restored data using the autoencoder. "Original" represents the original data, "encoded" represents the features extracted by the encoder, and "decoded" represents the original data reconstructed using a decoder based on those features. From this, it can be seen that there was a voice pattern for each data, and the input data was successfully restored through the autoencoder with an MSE loss value of about 0.00076. Therefore, it was found that the features extracted through the encoder of the autoencoder extracted an appropriate latent vector for data restoration, and the latent vector was used to perform user authentication and deep voice classification.

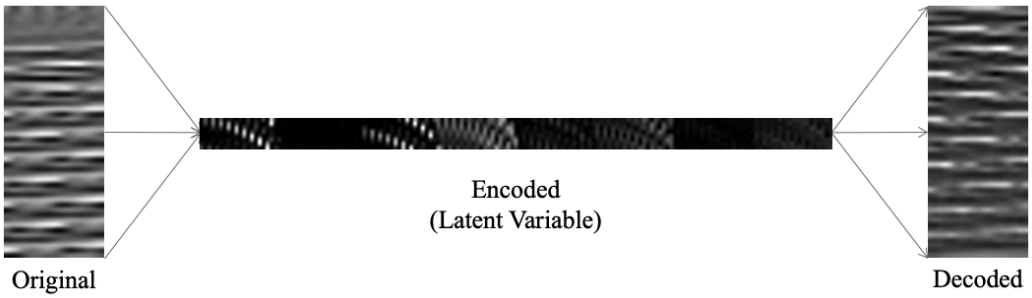

Figure 5. Original and decoded voice data samples.

4.3. Deep Voice Detection and User Authentication

We evaluated the performance of the model for user authentication and deep voice detection. As a performance indicator, the cross-entropy loss function was used in the training and validation process. In the testing process, performance was evaluated through the *F*1 *score* and *EER*. As a result of the experiment, the loss values achieved for the training and validation process were 0.021041 and 0.037903, respectively. Table 4 shows the *F*1 *scores* for user authentication and deep voice detection. In the training and validation process, the *F*1 *scores* achieved for user authentication were 99.03% and 98.55%, respectively, In the test process, the *F*1 *score* achieved was 99.05%. In the training and validation process, the *F*1 *scores* for deep voice detection were 99.98% and 99.95%, respectively. In the test process, deep voice detection achieved a score of 100%. Neither the loss value nor the *F*1 *score* dropped during the validation process, and a high *F*1 *score* was also achieved, indicating that overfitting did not occur during the test process. Accordingly, this shows that the model is suitable for deep voice detection and user authentication.

Table 4. *F*1 *scores* for user authentication and deep voice detection.

Method	Training	Validation	Test
User Authentication	99.03%	98.55%	99.05%
Deep Voice Detection	99.98%	99.95%	100%

Equal Error Rate for User Authentication

Table 5 shows the average *EER* for each class. The total average *EER* for user classification was 0.15. Therefore, we set the threshold to 0.15. In general, the default threshold was set as a random probability, but the proposed model was 3.75× higher than the random

probability of 0.04 when classifying 21 classes. In other words, since user authentication is possible only when classified with a high probability, the proposed model is robust in determining whether the sender is a deep voice or the correct sender.

Table 5. *EER* for user authentication in test dataset.

Deep Voice		User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9
0.1		0.14	0.11	0.09	0.08	0.14	0.25	0.07	0.05	0.3
User 10	User 11	User 12	User 13	User 14	User 15	User 16	User 17	User 18	User 19	User 20
0.08	0.1	0.2	0.12	0.04	0.15	0.18	0.12	0.13	0.21	0.35

4.4. Comparison with Previous Works

Table 6 shows a comparison with previous works using the ASVspoof2019 dataset mentioned in Section 2. Previous works were binary classifications that only detected real or fake voices for one user. Unlike previous works, in this work, we can simultaneously perform deep voice detection and user authentication through multi-class classification. To the best of our knowledge, this is the first system that can perform both user authentication and deep voice detection with ASVspoof2019 simultaneously.

In addition, most previous studies using various models achieved an EER value of 0.04. Our model achieved the EER values shown in Table 5 in Section 4.3, with an average EER of 0.15. However, in our system, real and fake classification as well as user authentication are possible. Therefore, a fair comparison is difficult, but as mentioned in Section 4.3, it has sufficient performance to perform deep voice detection and user authentication.

In other words, previous works involved checking whether a user was an authorized user or not, but this work involves a system that can authenticate which of several users he or she is. In addition, our system can deploy an autoencoder on the user's device to encode data in a separate environment. Through this, when extracting voice data in an inference situation, user authentication and deep voice detection are possible while preventing the original voice data from being exposed to the server. Therefore, it is possible to provide enhanced privacy protection by overcoming the vulnerability of data exposure in the previous deep voice detection technology.

Table 6. Comparison with previous works. O = provided method and X = not provided method.

Method	Deep Voice Detection	User Authentication	Privacy Preserved
This work	O	Classification for 20 users	O (by using encoded data)
Gomez et al. [15]	O	Real or fake for 1 user	X
Chen et al. [16]	O	Real or fake for 1 user	X
Wu et al. [18]	O	Real or fake for 1 user	X
Ma et al. [19]	O	Real or fake for 1 user	X
Wei et al. [20]	O	Real or fake for 1 user	X

5. Conclusions

In this paper, we proposed a system (i.e., DeepDetection) that simultaneously performs user authentication and deep voice detection to prevent voice phishing without privacy leakage. We designed a deep voice detection and user authentication model that achieves

high performance by preprocessing voice data using an autoencoder. In addition, the autoencoder can be deployed on the user's device for inference. Therefore, the problem of exposing voice data to the server can be prevented. The previous works involved detecting whether a user is a deep voice or an authenticated user, but this work involves a system that can authenticate which of several users he or she is while preserving privacy. To the best of our knowledge, this is the first system that can perform both user authentication and deep voice detection with ASVspoof2019 simultaneously. We achieved an F1 score of 100% in deep voice detection and an F1 score of 99.05% with an average EER of 0.15 in user authentication. This result means that user authentication is possible only when classified with high probability, so the proposed model is robust in determining whether the sender is a deep voice or a correct sender. Therefore, our system can effectively perform security-enhanced user authentication and deep voice detection, which is sufficient to prevent voice phishing. In future works, we will conduct additional research for a more robuste and safer voice phishing prevention system by creating human impersonation data.

Author Contributions: Conceptualization, Y.K.; Software, Y.K., W.K., S.L. and H.K.; Supervision, H.S.; Writing—original draft, Y.K.; Writing—review & editing, W.K., S.L. and H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00264, Research on Blockchain Security Technology for IoT Services, 100%).

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
2. Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics* **2021**, *10*, 2470. [CrossRef]
3. Ali, M.H.; Jaber, M.M.; Abd, S.K.; Rehman, A.; Awan, M.J.; Vitkutė-Adžgauskienė, D.; Damaševičius, R.; Bahaj, S.A. Harris Hawks Sparse Auto-Encoder Networks for Automatic Speech Recognition System. *Appl. Sci.* **2022**, *12*, 1091. [CrossRef]
4. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki Finland, 5–9 June 2008; pp. 1096–1103.
5. Delac, K.; Grgic, M. A survey of biometric recognition methods. In Proceedings of the Elmar-2004. 46th International Symposium on Electronics in Marine, Zadar, Croatia, 18 June 2004; pp. 184–193.
6. Naika, R. An overview of automatic speaker verification system. In *Intelligent Computing and Information and Communication*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 603–610.
7. Todisco, M.; Wang, X.; Vestman, V.; Sahidullah, M.; Delgado, H.; Nautsch, A.; Yamagishi, J.; Evans, N.; Kinnunen, T.; Lee, K.A. ASVspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv* **2019**, arXiv:1904.05441.
8. Wang, X.; Yamagishi, J.; Todisco, M.; Delgado, H.; Nautsch, A.; Evans, N.; Sahidullah, M.; Vestman, V.; Kinnunen, T.; Lee, K.A.; et al. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Comput. Speech Lang.* **2020**, *64*, 101114. [CrossRef]
9. Shen, J.; Pang, R.; Weiss, R.J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In Proceedings of the 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4779–4783.
10. Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.W.; Kavukcuoglu, K. WaveNet: A generative model for raw audio. *SSW* **2016**, *125*, 2.
11. AlBadawy, E.A.; Lyu, S.; Farid, H. Detecting AI-Synthesized Speech Using Bispectral Analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 104–109.
12. Wang, R.; Juefei-Xu, F.; Huang, Y.; Guo, Q.; Xie, X.; Ma, L.; Liu, Y. Deepsonar: Towards effective and robust detection of ai-synthesized fake voices. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1207–1216.
13. Ballesteros, D.M.; Rodriguez-Ortega, Y.; Renza, D.; Arce, G. Deep4SNet: Deep learning for fake speech classification. *Expert Syst. Appl.* **2021**, *184*, 115465. [CrossRef]
14. Lim, S.Y.; Chae, D.K.; Lee, S.C. Detecting Deepfake Voice Using Explainable Deep Learning Techniques. *Appl. Sci.* **2022**, *12*, 3926. [CrossRef]

15. Gomez-Alanis, A.; Peinado, A.M.; Gonzalez, J.A.; Gomez, A.M. A light convolutional GRU-RNN deep feature extractor for ASV spoofing detection. In Proceedings of the Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019, Volume 2019; pp. 1068–1072.
16. Chen, T.; Kumar, A.; Nagarsheth, P.; Sivaraman, G.; Khoury, E. Generalization of audio deepfake detection. In Proceedings of the Odyssey 2020 The Speaker and Language Recognition Workshop, Tokyo, Japan, 1–5 November 2020; pp. 132–137.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Wu, Z.; Das, R.K.; Yang, J.; Li, H. Light convolutional neural network with feature genuinization for detection of synthetic speech attacks. *arXiv* **2020**, arXiv:2009.09637.
19. Ma, H.; Yi, J.; Tao, J.; Bai, Y.; Tian, Z.; Wang, C. Continual learning for fake audio detection. *arXiv* **2021**, arXiv:2104.07286.
20. Wei, L.; Long, Y.; Wei, H.; Li, Y. New Acoustic Features for Synthetic and Replay Spoofing Attack Detection. *Symmetry* **2022**, *14*, 274. [CrossRef]
21. Wu, Z.; Li, H. Voice conversion versus speaker verification: An overview. In *APSIPA Transactions on Signal and Information Processing*; Cambridge University Press: Cambridge, UK, 2014; Volume 3.