IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Voice-Print Recognition System Using Python And Machine Learning With IBM Watson

[1]Prof Pritam Ahire, [2]Hariharan Achary, [3]Pratik Kalaskar, [4]Sourabh Shirke

[1]Professor, [2]Student, [3]Student, [4]Student
[1]Computer Engineerin,
[1]D.Y Patil Institute Of Engineering and Technology, Pune, India

Abstract: With the popularity of the various security system, identify the authentication of an exact user has been a key challenge for each system. The voiceprint recognition technology has some merit over another security system with flexible, economical terms. Therefore using voice as a key security tool will be useful for the ecosystem. In this system, we are going to use some machine learning concept SVM (Support Vector Machine). The SVC will be useful to differentiate the dataset and find out the actual required result so that user can get authentication to access the machine (Files, Folder, Documents), to make the system fully workable it is supported by Python on the frontend, with the help of the python SVC implement of it is successful and from the backend part, we are going to use IBM Watson which will provide actual power to the system. Using the IBM server APIs would be extra key to success to make the system more efficiently get the result as solving machine learning process on the client is hectic and hard to perform, that's why we are using the server of IBM which provides some free tools which are easily accessible and taking advantage of these tools we can easily achieve the required security and result for the system.

***Keywords* - Machine Learning, Python, IBM Watson Text to Speech, Fuzzy Wuzzy, Skitlearn, Support Vector Machine (SVM), Mel-Frequency Cepstral Coefficients.**

I. Introduction

In this advanced era, when people leverage social networking, online shopping, and online financial transactions without the need of being physically present at places. As a result, identity authentication has become the most critical security activity in the online world. The traditional solution uses a password or a private key (which is encrypted). In fact of this encryption, many people use easy passcode to remember "123456" to crack this kind of code are very easy for the intruder or any 3rd party person to hack out the system. The traditional solutions are a risky affair as the passwords are forgotten or lost and are also prone to hacker attacks. The best example would be the router password i.e admin and admin which can easily be attacked and all the connections to that device can be hacked out easily within that network. To overcome these drawbacks, we have decided voice security system. In the system, we have a key advantage that the person needs to be physically present to unlock the system and gain access. But a common question may arise that "I can record the person's voice and play it unlock the device, how it is secured?". The simple solution to its there is a difference playing sound and actual speaking which can be recognized with the SVC (Support Vector Classifier) algorithm and Machine learning which we are using SVM (Support Vector Machine). With the help of the python framework and using the machine learning concept of SVC (Support Vector Classifier) which will help us to distinguish the dataset and show us the required result with a higher accuracy rate. We have used MFCC (Mel-frequency cepstral coefficients), which plays a key role. Handling this huge dataset on the user side would be a hectic task, therefore to resolve this problem we are using the IBM Watson server. Using their APIs and processing the task on a higher powerful server would be an advantage for the system to work more efficiently. Combining both tools Python and IBM server we acquired a higher accuracy result.

II. Methodology

In this project, Python Framework and IBM Watson will be used. Python will be used for the frontend development side where all the functions and key components for the projects will be built. Which will process all the machine learning module and includes the functionality code for MFCC (Mel-Frequency cepstral coefficients). Which will help us to identify important components of an audio signal and train a model. This feature will be very beneficial for further processing over the analog signal of the audio. The input voice from the microphone will be extracted from the actual speech with less background noise and more clear result, the hamming technique to smooth the voice signal by the MFCC. The temporary voice is stored in the audio folder naming the file name as background_noise.wav. Using of Wav audio format is a benefit for us to record high fidelity with the accurate and lossless format. The IBM Watson Text to Speech service provides certain APIs which is beneficial for our project. These APIs will provide us backend power to analyze the data. The speech-synthesis capabilities of IBM to synthesize text into natural-sounding speech in a variety of languages, dialects (Specific Region Language), and voices. The service offers a customization interface that you can use to define sounds or translations for words. With the powerful server at the backend, we can implement and come out with a more precise result.

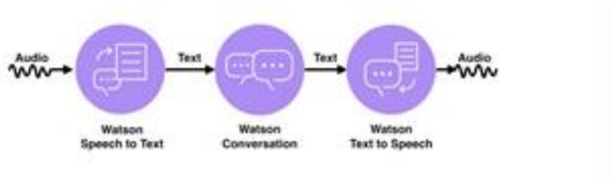

Fig. 1 - IBM Watson Flow.

III. Algorithm

3.1 SVC (Support Vector Classifier)

SVC algorithm is a supervised machine learning algorithm that can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs [15]. The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and maybe impractical beyond tens of thousands of samples. Considering large datasets using sklearn SVM LinearSVC or sklearn linear_model SGDClassifier.

$$A_{ij} = \begin{cases} 1, & if\ f(x) > 0\ for\ all\ x\ on\ the\ line\ segment\ connecting\ x_i and\ x_j \\ 0 & otherwise. \end{cases}$$

The further explanation of the SVC algorithm can be easily understood with the help of the diagram and its structure. The following figure will explain the actual machine learning SVC used in the voice-print security system.

Condition 1:

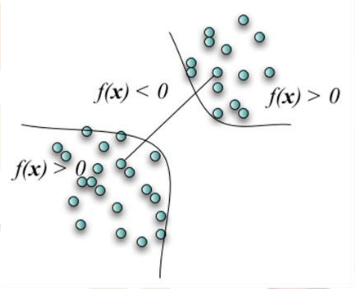

Fig. 2 - The line segment that connects points in different clusters has to go through a low-density region in data space where the SVDD returns a negative value.

Left f(x) > 0 is a dataset and Right f(x) > 0 is another dataset in which are easily identical and can be separate out with f(x) < 0. This is the most simple and easy test case we can get the result at a single glance. Not much cores or process is not required to figure out the solutions, But it won't be a similar case.

Condition 2:

When the mixed dataset occurs the complexity of the problem increases, eg of mixed data is shown in Fig (3). The machine needs more cores and more data to analyze it and figure out the accurate result [15]. To solve such kind of complex problem we can increase the number of hidden layers in SVM (Scaler Vector Machine).

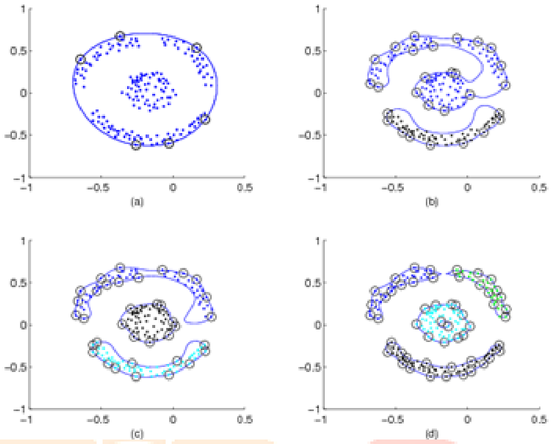

Fig 3 - Contours generated by SVDD as γ is increased.

We can use 2 hidden layers, which 1st layer will have 5 neurons present init and the 2nd layer will have 2 neurons which will help out the output to varying weights and differentiate the datasets. The below Fig (4) shows the result when the dataset is differentiated with the help of the SVC algorithm.

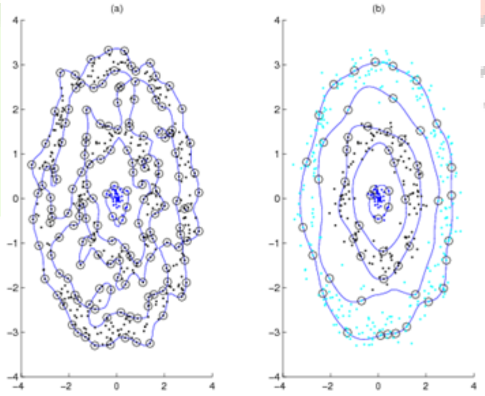

Fig 4 - Allowing for outliers allows SVC to separate noisy data

3.2 MLP Classifier

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(X) = R^m \rightarrow R^0$ by work through data training, where m is the input value is and the 0 is output size. Given a set of features $X = x1, x2, . . . , x_m$ and references γ, it can read a non-linear function scale with a split or a reversal. It differs from ordering in order, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers [17]. Figure 5 shows one hidden layer of MLP by scale extraction.

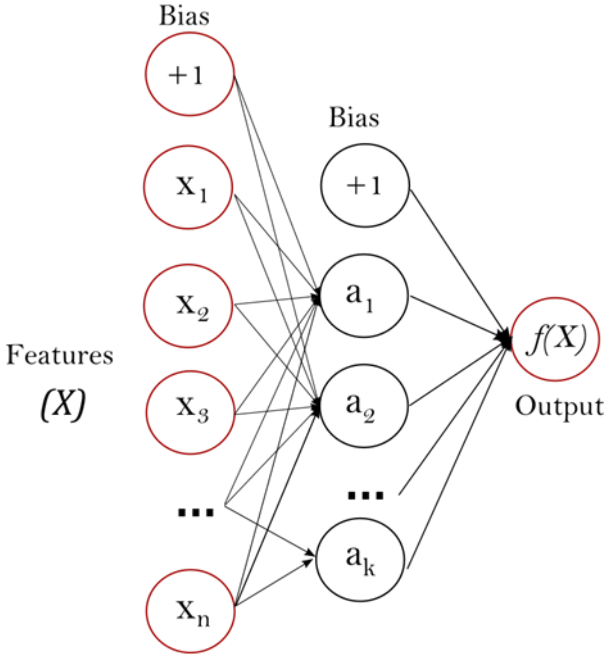

Fig 5 - Hidden Layer of MLP

The left layer, known as the input layer, contains a set of neurons $\{x_i i | x_1, x_2, \ldots, x_m\}$ representing the input features. Each neuron in the hidden layer converts values from the previous layer by a cut line $\omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_m x_m$, followed by indirect activation of $g(.): R \rightarrow R$ as a hyperbolic tan function. The output layer detects values from the last hidden layer and converts them into output values [20]. The module contains social attributes breaks coefficients & intercepts. Coefficients list of weight matrices, where the weight matrix i index represents the weights between layer i and layer i + 1. Intercepts list of bias vectors, where the vector in the code represents the bias values included in the i + 1.

.

IV. ARCHITECHTURE

The MFCC feature extraction technique includes windowing the signal, applying the DFT, taking the log of the magnitude, and then warping the frequencies on a Mel scale, followed by applying the inverse DCT [18]. The detailed description of various steps involved in the MFCC feature extraction is explained below.

MFCC Feature

4.1 Pre-emphasis*:* Pre-emphasis refers to filtering that emphasizes the higher frequencies. Its purpose is to balance the spectrum of voiced sounds that have a steep roll-off in the high-frequency region [13]. For voiced sounds, the glottal source has an approximately −12 dB/octave slope. However, when the acoustic energy radiates from the lips, this causes a roughly +6 dB/octave boost to the spectrum [1]. As a result, a speech signal when recorded with a microphone from a distance has approximately a −6 dB/octave slope downward compared to the true spectrum of the vocal tract. Therefore, pre-emphasis removes some of the glottal effects from the vocal tract parameters. The most commonly used pre-emphasis filter is given by the following transfer function Where the value of *b* controls the slope of the filter and is usually between 0.4 and 1.0.

$$H(z) = 1 - bz^{-1}$$

4.2 Frame blocking and windowing*:* The speech signal is a slowly time-varying signal. For stable acoustic characteristics, speech needs to be examined over a sufficiently short time. Therefore, speech analysis must always be carried out on short segments across which the speech signal is assumed to be stationary. Short-term spectral measurements are typically carried out over 20ms windows, and advanced every 10ms [2],[3]. Advancing the time window every 10ms enables the temporal characteristics of individual speech sound to be tracked, and the 20ms analysis window is usually sufficient to provide good Spectral resolution of these sounds and at the same time short enough to resolve significant temporal characteristics. The purpose of the overlapping analysis is that each speech sound of the input sequence would be approximately centered at some frame. On each frame, a window is applied to taper the signal towards the frame boundaries. Generally, Hanning or Hamming windows are used. This is done to enhance the harmonics, smooth the edges, and reduce the edge effect while taking the DFT on the signal [1]. For the continuity of the start and the end, the Hamming window as in the equation is used.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N+1}\right) \quad 0 \leq n \leq N$$

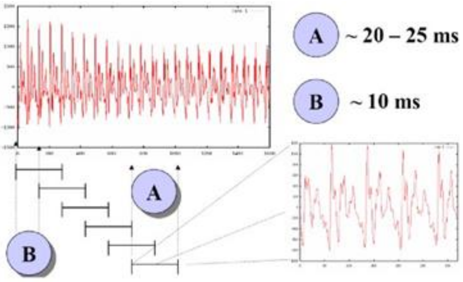

Image from Bryan Pellom

Fig 6 - Framing & Windowing

4.3 DFT spectrum: Each windowed frame is converted into a magnitude spectrum by applying DFT.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi nk}{N}};$$

$$0 \le k \le N-1$$

4.4 Mel spectrum: Mel spectrum is computed by passing the Fourier transformed signal through a set of band-pass filters known as Mel-filter bank. A Mel is a unit of measure based on the human ear's perceived frequency. It does not correspond linearly to the physical frequency of the tone, as the human auditory system does not perceive pitch linearly. The Mel scale is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz [4], the approximation of Mel from physical frequency can be expressed as Where f denotes the physical frequency in Hz, and f_{Mel} denotes the perceived frequency [2].

$$f_{Mel} = 2595\, log_{10}\left(1 + \frac{f}{700}\right)$$

Filter banks can be implemented in both the time domain and frequency domain. For MFCC computation, filter banks are generally implemented in the frequency domain. The center frequencies of the filters are normally evenly spaced on the frequency axis [16]. However, to mimic the human ears perception, the warped axis, according to the nonlinear function given in Equation, is implemented. The most commonly used filter shaper is triangular, and in some cases, the Hanning filter can be found [1]. The triangular filter banks with Mel frequency warping are given in Fig (7) The Mel spectrum of the magnitude spectrum *X (k)* is computed by multiplying the magnitude spectrum by each of the triangular Mel weighting filters [5],[6].

$$s(m) = \sum_{k=0}^{N-1} [|X(k)|^2 H_m(k)];$$

$$0 \le m \le M-1$$

where *M* is the total number of triangular Mel weighting filters. *Hm (k)* is the weight given to the k^{th} energy spectrum bin contributing to the m^{th} output band and is expressed as below with *m* ranging from 0 to M-1.

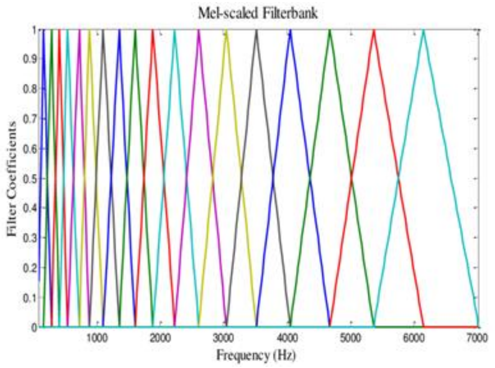

Fig 7 - Mel Filter Bank

$$H(m) = \begin{cases} \mathbf{0}, & k < f(m-1) \\ \dfrac{2\big(k - f(m-1)\big)}{f(m) - f(m-1)}, & f(m-1) \le k \le f(m) \\ \dfrac{2\big(f(m+1) - k\big)}{f(m+1) - f(m)}, & f(m) < k \le f(m+1) \\ \mathbf{0} & k > f(m+1) \end{cases}$$

4.5 Discrete Cosine Transform (DCT): Since the vocal tract is smooth, the energy levels in adjacent bands tend to be correlated. The DCT is applied to the transformed Mel frequency coefficients produces a set of cepstral coefficients. Before computing DCT, the Mel spectrum is usually represented on a log scale. This results in a signal in the cepstral domain with a quefrequency peak corresponding to the pitch of the signal and several formants representing low quefrequency peaks [1]. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring or truncating higher order DCT components. Finally, MFCC is calculated as [1],[13]

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

$$n = 0, 1, 2, . \; . \; . \; . \; , C - 1$$

where *c(n)* is the cepstral coefficients, and *C* is the number of MFCCs. Traditional MFCC systems use only 8–13 cepstral coefficients. The zeroth coefficient is often excluded since it represents the average log-energy of the input signal, which only carries little speaker-specific information.

4.6 Dynamic MFCC features: The cepstral coefficients are usually referred to as static features since they only contain information from a given frame. The extra information about the temporal dynamics of the signal is obtained by computing the first and second derivatives of cepstral coefficients [7],[8],[9]. The first-order derivative is called delta coefficients, and the second-order derivative is called delta-delta coefficients. Delta coefficients tell about the speech rate, and delta-delta coefficients provide information similar to the acceleration of speech. The commonly used definition for computing dynamic parameter is [7]

$$\triangle C_m(n) = \frac{\sum_{i=-T}^{T} k i C_m(n+i)}{\sum_{i=-T}^{T} |i|}$$

where *cm(n)* denotes the *m*th feature for the *nth* time frame, k_i is the *i*th weight, and *T* is the number of successive frames used for computation. Generally, *T* is taken as 2. The delta-delta coefficients are computed by taking the first-order derivative of the delta coefficients.

V. Authentication Method

The two-layer Authentication method includes two processes: one for registration, one for user authentication.

5.1 The registration process: Before a user can work with the system, a user must first register his voice. The program is recording the user's voice and sends the recording back-end. Next, Authentication Speaker Verification (ASV) system detects file recording and output a user voice model (e.g., voiceprint) [19]. Finally, the ASV system saves the user's voice model into the database.

5.2 Authentication process: The authentication process will do tested in two, different perspectives: user and backend.

From a user's point of view, the authentication process consists of actions:

1) Trigger voice assistant.

2) Speak service command. User design the flow can add a step to the end, so as not to disturb the real user flow for voice assistants.

When the voice listener, users may speak his/her password to unlock the data or files, then wait for the system to run or reject the request. From a backend perspective, the process consists of six actions. While the user speaks the service command, the backend records his voice.

In the Next Step, the backend sends the recording to the speaker the verification process after the user has completed the command. The speaker authentication system will then determine whether a voiceprint recording is similar to a registered user. When he finds out the similarity, the program identifies the speaker as a specific user. If the speaker verification system cannot find the best match model among registered users, identification failed. The back will disconnect and tell the user that the identification failed. The user has 5 attempts if it fails, the authentication process will be terminated.

In the next step, if the diagnosis is successful the data or file is provided to the user. The system is called two-layer authentication because uses dual voice input to identify and verify the user. This method prevents unauthorized speaker's access to the data. The attacks mentioned earlier will not cross this layer unless the attackers can get voice recordings from registered users. If the attacker you can get the voice recording of the registered user, and pass it for the first time layer.

VI. Conclusion

In this paper, we try to present a security system using voice biometrics with the help of machine learning and python on the frontend side. The trained data with SVC (Support Vector Classifier). As we all voice security has a low accuracy rate but using this machine learning technique we can increase this accuracy rate. The main idea of implementing this system is to provide more security to the system. With the help of SVC, MFCC, and MLP Classifier with powered backend IBM Watson. The background noise can be removed with python packages. This package will help us to have a clear output that can be compared with the existing dataset and if the output and present data are matched then we can get access to the system.

However in the case of complex background noise, the method of voiceprint identification will be affected a little bit, but in the future, this complicated situation can be studied and solved with more advanced machine learning concepts.

VII Acknowledgement

This paper was successful with the contribution of many people who were directly or indirectly connected to this paper as some concepts can understand with help of the internet and other authors, who had broken-down the concept which made it easy to this paper. I would like to express our gratitude to Prof. Pritam Ahire for their help and advice. He provided many favorable suggestions for the grammar and experimental parts of machine learning concepts.

VIII Acknowledgement

[1] J.W. Picone, Signal modeling techniques in speech recognition. Proc. IEEE **81**, 1215–1247 (1993)

[2] J.R. Deller, J.H. Hansen, J.G. Proakis, *Discrete-Time Processing of Speech Signals* (Prentice Hall, NJ, 1993)

[3] J. Benesty, M.M. Sondhi, Y.A. Huang, *Handbook of Speech Processing* (Springer, New York, 2008)

[4] J. Volkmann, S. Stevens, E. Newman, A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. **8**, 185–190 (1937)

[5] Z. Fang, Z. Guoliang, S. Zhanjiang, Comparison of different implementations of MFCC. J. Comput.Sci. Technol. **16**, 582–589 (2000)

[6] G.K.T. Ganchev, N. Fakotakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in *Proceedings of International Conference on Speech and Computer(SPECOM)* (2005), pp. 191–194

[7] L. Rabiner, B.-H. Juang, B. Yegnanarayana, *Fundamentals of Speech Recognition* (Pearson Education,

London, 2008)

[8] S. Furui, Comparison of speaker recognition methods using statistical features and dynamic features IEEE Trans. Acoust. Speech Sig. Proc. **29**, 342–350 (1981)

[9] J.S. Mason, X. Zhang, Velocity and acceleration features in speaker recognition, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (1991), pp. 3673–3676

[10] Patricia Melin, JericaUrias, Daniel Solano, Miguel Soto, Miguel Lopez, and Oscar Castillo," Voice Recognition with Neural Networks, Type-2 Fuzzy Logic, and Genetic Algorithms", Engineering Letters, 13:2, EL_13_2_9,4 August 2006.

[11] M. S. Likitha, S. R. R. Gupta, K. Hasitha and A. U. Raju, "Speech-based human emotion recognition using MFCC", *2017 International Conference on WirelessCommunications Signal Processing and Networking(WiSPNET)*, vol. 4, no. 5, pp. 2257-2260, 2017.

[12] D. D. Joshi and M. B Zalte, "Speech emotion recognition: a review", *IOSR J. Electron. Commun. Eng. (IOSR-JECE)*, vol. 4, no. 4, pp. 34-37, 2013.

[13] W. Han, C. F. Chan, C. S. Choy, and K. P. Pun, "An efficient MFCC extraction method in speech recognition", *2006 IEEE international symposium on circuits and systems*, vol. 5, no. 2, pp. 4, 2006.

[14] A. J. Harris and D. C. Yen, "Biometric authentication", *assuring access to information Information Management & Security 10/1*, pp. 12-19, 2002.

[15] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, DOI: 10.1109/5254.708428.

[16] Bhadragiri Jagan Mohan and Ramesh Babu N., "Speech recognition using MFCC and DTW," 2014 International Conference on Advances in Electrical Engineering (ICAEE), Vellore, India, 2014, pp. 1-4, DOI: 10.1109/ICAEE.2014.6838564.

[17] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," in IEEE Transactions on Neural Networks, vol. 3, no. 5, pp. 683-697, Sept. 1992, DOI: 10.1109/72.159058.

[18] Z. Wanli and L. Guoxin, "The research of feature extraction based on MFCC for speaker recognition," Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, Dalian, China, 2013, pp. 1074-1077, DOI: 10.1109/ICCSNT.2013.6967289.

[19] Y. Chang and M. J. Dupuis, "My Voiceprint Is My Authenticator: A Two-Layer Authentication Approach Using Voiceprint for Voice Assistants," 2019 IEEE SmartWorlda, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 2019, pp. 1318-1325, DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00243.

[20] A. C. Morris, Dalei Wu, and J. Koreman, "MLP trained to separate problem speakers provide improved features for speaker identification," Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology, Las Palmas, Spain, 2005, pp. 325-328, DOI: 10.1109/CCST.2005.1594867.