This is a preprint of:
Morrison G.S., Weber P., Enzinger E., Labrador B., Lozano-Díez A., Ramos D., González-Rodríguez J. (2022). Forensic voice comparison: Human-supervised-automatic approach. In Houck M., Wilson L., Lewis S., Eldridge H., Lothridge K., Reedy P. (Eds.), Encyclopedia of Forensic Sciences (3rd Ed.), vol. 2, pp. 720–736. Elsevier. https://doi.org/10.1016/B978-0-12-823677-2.00182-3
Preprint at http://forensic-voice-comparison.net/encyclopedia/

Forensic voice comparison – Human-supervised-automatic approach

Authors

Geoffrey Stewart Morrison

Forensic Data Science Laboratory, Aston University
Forensic Evaluation Ltd
geoff-morrison@forensic-evaluation.net

Philip Weber

Forensic Data Science Laboratory, Aston University
p.weber1@aston.ac.uk

Ewald Enzinger

Forensic Data Science Laboratory, Aston University
Eduworks Corporation
ewald-enzinger@forensic-evaluation.net

Beltrán Labrador

AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid
beltran.labrador@uam.es

Alicia Lozano-Díez

AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

alicia.lozano@uam.es

Daniel Ramos

AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

daniel.ramos@uam.es

Joaquín González-Rodríguez

AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

joaquin.gonzalez@uam.es

Keywords

[10–15 keywords, listed alphabetically]

automatic speaker recognition

calibration

forensic speaker comparison

forensic speaker identification

forensic speaker recognition

forensic speech science

forensic voice comparison

likelihood ratio

validation

x-vector

Abstract

[50–100 words]

The human-supervised-automatic analytical approach to forensic voice comparison in conjunction with the likelihood-ratio interpretive framework is described. Practitioner tasks are described, including adoption of relevant hypotheses for the case, assessment of the conditions of the questioned-speaker and known-speaker recordings in the case, and selection of data representing the relevant population and reflecting the conditions for the case. Software tools are also described. An example is provided of a forensic-voice-comparison system based on state-of-the-art automatic-speaker-recognition technology. Also described are the calibration and validation of that system using a benchmark dataset reflecting the conditions of a real forensic case.

Key points

[short bulleted list of key points]

- likelihood-ratio framework
    - hypotheses
    - relevant population
    - common-source likelihood-ratio model
    - recording conditions
    - training and validation data
- software tools based on state-of-the-art automatic-speaker-recognition technology
    - diarization and voice activity detection

  - feature extraction
  - x-vector extraction
  - dimension reduction and mismatch compensation
  - calculation of uncalibrated likelihood ratio
  - calibration
- validation

Acknowledgements

The writing of this entry was supported by Research England’s Expanding Excellence in England Fund as part of funding for the Aston Institute for Forensic Linguistics 2019–2023.

1 Introduction

The reader is assumed to already have some familiarity with the following topics, to the level which may be found in general introductions to forensic voice comparison: analytical approaches to forensic voice comparison, including the human-supervised-automatic approach; interpretive frameworks applied to forensic voice comparison, including the likelihood-ratio framework; and validation of forensic-evaluation systems, including validation of systems that output likelihood ratios.

The human-supervised-automatic analytical approach combined with the likelihood-ratio interpretive framework is described in greater detail than is usual in general introductions to forensic voice comparison. Recent reviews of automatic-speaker-recognition technology include Lee et al. (2020), Matějka et al. (2020), and Villalba et al. (2020). Morrison et al. (2020) provides an overview of the application of automatic-speaker-recognition technology to forensic voice comparison. Application of more recent automatic-speaker-recognition technology is described in Weber et al. (2022a, 2022b).

For concreteness, in the current entry an example human-supervised-automatic forensic-voice-comparison system is described. The example is based on an alpha version of the core software tools of the E^3 Forensic Speech Science System (E^3FS^3). E^3FS^3 is being developed by the Forensic Data Science Laboratory at Aston University, with contributions from AUDIAS – Audio, Data Intelligence and Speech at Universidad Autónoma de Madrid, and with additional contributions from multiple other research laboratories and operational forensic laboratories. E^3FS^3 is being developed for both research and casework use, and includes open-code software tools. The E^3FS^3 software tools are designed to be flexible and provide the user with various options. For simplicity, the example focuses on a single set of options. Although details may vary, the example system is broadly similar to other state-of-the-art systems. Also for concreteness, the discussion of protocols is based on those used by Forensic Evaluation Ltd. Protocols used by other forensic-service providers may vary.

Descriptions are provided of:

- hypotheses and data, including:
    - the adoption of the relevant hypotheses for a case,
    - the assessment of the conditions of the questioned-speaker and known-speaker recordings in the case, and
    - the selection of data representing the relevant population and reflecting the conditions for the case
- the core software tools of the example system
- a benchmark validation of the example system

2 Hypotheses and data

2.1 Hypotheses

State-of-the-art human-supervised-automatic forensic-voice-comparison systems calculate common-source likelihood ratios for which the same-speaker versus different-speaker hypotheses are:

$\boldsymbol{H_1}$**:** the speakers on the questioned-speaker and the known-speaker recordings are the same speaker

versus

$\boldsymbol{H_2}$**:** the speakers on the questioned-speaker and the known-speaker recordings are not the same speaker but two different speakers each selected at random from the relevant population

2.2 Relevant population

The relevant population is the population from which the questioned speaker could potentially have come if they were not the known speaker. The relevant population can usually be restricted to either male or female speakers who speak a particular language with a particular accent (Morrison et al., 2016). By listening, it is usually clear to non-experts such as judges and jury members whether the speaker on the questioned-speaker recording is male or female, what language they are speaking, and broadly what accent of that language they are speaking. This, however, is not always the case, for example, it may be unclear whether the speaker on the questioned-speaker recording is male or female. If any of these things are disputed, then they may become issues requiring forensic evaluation. Usually, a forensic practitioner, in consultation with the instructing party, can define a relevant population to adopt for the case. In their casework report, the forensic practitioner should clearly state what population they have adopted, and they may request that the instructing party specify the relevant population in their letter of instruction. The choice of relevant population is a subjective judgment, and it can potentially be disputed. If the judge at an admissibility hearing or the judge or jury during trial is not convinced that the population adopted is a reasonable relevant population for the case, then the likelihood ratio that the forensic practitioner calculates will not be meaningful for the case – it will be answering a question about a different population than the one that the judge or jury considers relevant for the case.

2.3 Common-source likelihood-ratio model

In general, a common-source likelihood-ratio model has the form given in Equation (1) in which Λ is the likelihood ratio, $f\left(x_{\mathrm{q}}, x_{\mathrm{k}} \middle| M\right)$ is a joint probability-density function, x_{q} and x_{k} are feature vectors characterizing the speech of the speaker of interest on the questioned-speaker and known-speaker recordings respectively (in our example system these are x-vectors), and M_{s} and M_{d} are the same-speaker and different-speaker models respectively.

(1)

$$\Lambda = \frac{f\left(x_{\mathrm{q}}, x_{\mathrm{k}} \middle| M_{\mathrm{s}}\right)}{f\left(x_{\mathrm{q}} \middle| M_{\mathrm{d}}\right) f\left(x_{\mathrm{k}} \middle| M_{\mathrm{d}}\right)}$$

In order to train (or adapt) the statistical models that calculate the likelihood ratio, the practitioner must use data from recordings of speakers sampled from the population that has been adopted as the relevant population for the case. If the data are not sufficiently representative of the relevant population for the case, then the likelihood ratio calculated will answer a different question than the one defined by the stated hypotheses. In their report, the practitioner should describe the data that they use for training. Whether the data are sufficiently representative of the relevant population is a subjective judgment, and it can potentially be disputed. Likewise, data used for validating the system must be sufficiently representative of the relevant population for the case. Usually, a single dataset intended to be representative of the relevant population will be obtained or selected, and that dataset will then be divided into a training set and a validation set. If these data are not actually representative of the relevant population, no amount of empirical testing will reveal this (Morrison, 2021).

2.4 Recording conditions

In addition to the data used for training and validation being sufficiently representative of the relevant population for the case, they must also be sufficiently reflective of the conditions of the questioned-speaker and known-speaker recordings in the case. Hansen & Bořil (2018) provide a taxonomy of sources of speaker-intrinsic and speaker-extrinsic variability. Common speaker-intrinsic conditions or speaking styles include normal vocal effort and raised vocal effort. Slight to moderate raised vocal effort often occurs when a speaker is in a noisy environment or is communicating over a poor-quality telecommunications channel. Shouting and whispering are obvious extreme examples of speaking styles. More extreme speaking styles, such as whispering, have more negative effects on the performance of automatic-speaker-recognition technology than do less extreme speaking styles, such as slight to moderate raised vocal effort (Kelly & Hansen, 2021). Speaking one language on one recording and another language on another recording would also be a change in speaking style. A speaker's speech varies from occasion to occasion, and variation tends to be greater across longer time intervals between recordings. Very long time intervals between when the questioned-speaker and known-speaker recordings were made should be accounted for in the statistical modeling process (Morrison & Kelly, 2019). Common speaker-extrinsic conditions include different types and volumes of background noise, different amounts of reverberation, different distances from the speaker to the microphone, transmission of the speech signal through telecommunications systems that include bandpass filters and lossy compression, and recordings being saved using lossy compression. The length of the speech of the speaker of interest on a recording is also a condition. If high-quality recordings of speakers in suitable speaker-intrinsic conditions are available, speaker-extrinsic conditions can potentially be simulated by adding noise, convolving the audio signal with filters, and compressing and decompressing the signal. Data from longer recordings can easily be truncated to reflect shorter recordings.

Assessing speaker-intrinsic conditions will usually require listening to the questioned-speaker and known-speaker recordings (see comments on listening in the Diarization and VAD subsection below). Some speaker-extrinsic conditions such as signal-to-noise ratio can be quantitatively analyzed. More sophisticated channel-characterization tools may provide additional information such as information relating to what codecs may have been applied to a recording. Through the instructing party, the practitioner should also make enquiries as to the technical properties of systems used to make the casework recordings, e.g., a recording may be received as "pulse code modulation" (PCM, the standard uncompressed encoding for audio recordings), but it may have originally been saved in a lossy format and then exported to PCM. The instructing party will not usually have such technical information to hand, and the enquiry will usually have to be passed on to the person or team responsible for installing and maintaining the recording equipment at the organization that supplied the recording.

2.5 Training and validation data

There are often mismatches between the conditions of the questioned-speaker recording and the conditions of the known-speaker recording. For each speaker in the training and validation datasets, there should be at least one recording that reflects the conditions of the questioned-speaker recording and at least one recording that reflects the conditions of the known-speaker recording. In their report, the practitioner should describe the conditions of the questioned-speaker and known-speaker recordings for the case, and how they obtained, selected, or simulated recordings that reflect the conditions for the case. The choice of training and validation data is a subjective judgment, and whether the data are sufficiently reflective of the conditions of the case can potentially be disputed.

The relevant population and the conditions of the questioned-speaker and known-speaker recordings can be highly variable from case to case. State-of-the-art automatic-speaker-recognition technology can produce good results over a range of different conditions, but the major impediment to conducting forensic casework is availability of training and validation data representing the populations and reflecting the conditions of specific cases. Sometimes, new recordings can be made representing the population and reflecting the conditions for a specific case, but this is usually not practical – it depends on how easy it is to collect data for the specific population and specific set of conditions, the time available, and the budget available. Substantial investment is needed to build databases covering populations and conditions that are anticipated to occur in a substantial proportion of future cases. If the number of such databases increases and they cover a wider range of populations and conditions, then it will become practical to perform evaluations in a larger proportion of the cases for which forensic voice comparison is requested.

Research is needed to assess the effects of varying speaker-intrinsic and speaker-extrinsic conditions, and of varying population. If changes in some conditions result in substantial changes in likelihood-ratio output, then this will inform practitioners that in future casework they should use data that closely reflect those conditions and not substitute data in one condition for data in another. If changes in some conditions result in negligible changes in likelihood-ratio output, then this will inform practitioners that in future casework they can substitute data in one condition for data in another, thus making it easier to obtain data that are sufficiently reflective of the conditions for a case. The results of such research will also inform future data-collection efforts.

The poorer the quality and the shorter the duration of casework recordings, and the greater the mismatch in recording conditions between questioned-speaker and known-speaker recordings, the poorer the performance of the forensic-voice-comparison system is expected to be. In principle, however, there is no minimum threshold below which forensic voice comparison cannot be performed. As long as suitable training and validation data can be obtained, the system can be trained and validated under conditions reflecting those of the case. A decision can subsequently be made about whether the performance of the system under those conditions is good enough to proceed to use it to compare the questioned-speaker and known-speaker recordings for the case. In practice, before training and validating the system, based on existing research and validation literature, the practitioner may be able to advise the instructing party in broad terms about the expected level of performance. If the level of performance is expected to be poor, a decision may be made to not proceed with training and validation (and not to proceed with data collection if it would be needed for training and validation).

3 Software tools

3.1 System architecture

The high-level architecture of the example system's core software tools is presented in

Figure 1. It consists of the following stages:

1. speaker diarization and voice-activity detection (VAD)
2. feature extraction
3. x-vector extraction
4. dimension reduction and mismatch compensation using linear discriminant analysis (LDA)
5. calculation of uncalibrated likelihood ratios (scores) using probabilistic linear discriminant analysis (PLDA)
6. calibration

<Figure 1 about here>

Figure 1. High-level architecture for the example human-supervised-automatic forensic-voice-comparison system's core software tools.

Data from the questioned-speaker recording and data from the known-speaker recording are processed in parallel through Stages 1–4. Stages 5 and 6 operate on data from pairs of recordings. Recordings used for training and validating the system (not shown in Figure 1) are processed in the same manner as the data from the questioned-speaker and known-speaker recordings. Terminologically: Stage 1 (diarization and VAD) are key parts of preprocessing; Stage 3 (x-vector extraction) constitutes the frontend model; and Stages 4–6 (LDA, PLDA, and calibration) constitute the backend models.

We describe each stage of the system in its own subsection below.

3.2 Diarization and VAD

Audio recordings received for forensic evaluation often include the speech of more than one speaker on the same recording channel. Speaker diarization is the process of dividing a recording into sections spoken by different speakers. Usually, only one speaker on a recording is of interest. In this situation, the speaker-diarization task is to find the sections of the recording that correspond to speech of the speaker of interest. Usually, the different speakers on a recording sound sufficiently different from each other that this is a trivial task for a forensic practitioner to perform manually. If this is not the case, then the identity of the speaker at various points in the recording may be a question requiring forensic evaluation. For manual diarization, the practitioner uses a software tool that visually presents the waveform, allows the practitioner to select and listen to sections of the recording, and allows them to add markers and labels indicating the sections that contain speech of the speaker of interest. The practitioner should listen in a quiet environment using reference headphones. Marking and labeling can be performed using any one of multiple commercial or freeware software tools designed for general use, e.g., Audition, Sound Forge, Audacity, or Praat. The example system includes SoundLabeller, a marking and labeling tool that is designed specifically for this task.

A protocol can be adopted whereby one practitioner performs the diarization task, a second practitioner checks the results, and a pre-specified process to resolve any disagreements is then used. A protocol to reduce the potential for cognitive bias is to have one practitioner diarize the questioned-speaker recording and another practitioner diarize the known-speaker recording. That way, no individual practitioner auditorily compares the questioned-speaker and known-speaker recordings, so no practitioner can form a subjective judgment as to whether the two recordings are recordings of the same speaker or not. Strictly following both protocols would require a total of four practitioners. If this is impractical, a compromise would be to have a long time interval (e.g., at least several weeks) between when any individual practitioner (e.g., the checker) listens to the questioned-speaker recordings and the known-speaker recordings.

Practitioners may manually diarize the questioned-speaker and known-speaker recordings for a case, but this may be impractical if there are a large number of recordings or if the recordings are very long. Manual diarization is unlikely to be practical for the large numbers of recordings used for training and validating the forensic-voice-comparison system. Automatic diarization is itself a form of automatic speaker recognition. The example system uses the automatic-diarization method that performed the best in the DIHARD'19 diarization challenge, the VBx algorithm (Diez et al., 2019, 2020a; Landini et al., 2020, 2022). (All the data for the benchmark validation described below were supplied already diarized.)

VAD is the detection and selection of sections of the recording that contain speech, as opposed to silence, background noise only, or transient noises. VAD is a prerequisite for diarization, but it is also required to find the sections of a recording containing speech when there is only one speaker on a recording channel. Although VAD could be performed manually, automatic VAD is preferred in order to obtain consistent results. If a practitioner performs manual diarization, they should mark the start of a section of speech a little early and the end a little late. Automatic VAD will then further truncate the manually marked section.

Simple automatic VAD methods are based only on the intensity of the signal in the recording, but these methods do not perform well when there is background noise on the recording, as is common in casework recordings. More sophisticated automatic VAD methods attempt to distinguish speech sounds from non-speech sounds. Supervised methods require labeled training data, and tend to not perform well if they are applied to recordings whose conditions differ from those of the training data. Unsupervised methods do not require labeled training data, and are more robust to changes in conditions. Unsupervised methods can achieve similar levels of performance to supervised methods when the latter are trained and tested on the same conditions (Kinnunen et al., 2016; Nautsch et al., 2016; Tan et al., 2020).

The example system uses the rVAD-fast algorithm (Tan et al., 2020). This algorithm applies two noise-removal processes: The first process attempts to remove transient noises, and the second process attempts to remove background noise (this is background-noise removal for the purpose of performing VAD, features used for forensic voice comparison are extracted from the unmodified audio signal). The next stage in the algorithm searches for voiced speech sounds using a spectral flatness detector (which is faster than fundamental-frequency detection, which was used in the original rVAD algorithm). In order to also include voiceless sounds, the sections of the recording identified as containing voiced sounds are extended both before and after by 60 frames (600 ms). The final stage uses heuristics based on the energy differences between frames to select frames deemed to be speech.

3.3 Feature extraction

Until recently, "mel-frequency cepstral coefficients" (MFCCs; Davis & Mermelstein, 1980) were the most commonly used features for automatic-speaker-recognition systems, but "log-mel-filterbank features" have been found to be more effective for x-vector systems (Alam et al., 2020; Landini et al., 2020; Lee et al., 2020). The example system uses the implementation of log mel filterbanks described in Young et al. (2015, §3.1.5).

The steps for extracting log-mel-filterbank features are described below, see also Figure 2 in which the numbers correspond to the numbered steps below.

1. The speech signal is multiplied by a bell-shaped window. In our example system, this is a Hamming window with a duration of 25 ms, i.e., 200 samples if the sampling frequency of the recording is 8 kHz.

2. The power spectrum of the windowed signal is calculated using a discrete Fourier transform (DFT). The power spectrum consists of the squares of the magnitudes

of the components of the Fourier series (phase information is discarded). For computational efficiency, the example system uses a 512 point fast Fourier transform.

3. The power spectrum is multiplied by each filter in a filterbank. These are a series of triangular shaped filters that are equally spaced in the mel-frequency scale. The example system uses 40 filters that together cover the frequency range 0–4 kHz. Each filter has a 50% overlap with each of its neighbors.

4. For each filter in the filterbank, the logarithm is taken of the result of multiplying the power spectrum by that filter. This creates a set of 40 values that are output as a vector of log-mel-filterbank features.

The window is advanced through the recording. In the example system, it is advanced by 10 ms (80 samples). Each time-interval covered by a window is called a frame, and in the example system there is a 60% overlap between adjacent frames. Steps 1 through 4 are then repeated to produce another vector of log-mel-filterbank features. The window is repeatedly advanced until feature vectors have been extracted from all sections of the recording corresponding to the speech of the speaker of interest. A series of feature vectors arranged in chronological order will be referred to as a feature matrix.

<Figure 2 about here>

Figure 2. Procedure for the calculation of log-mel-filterbank feature vectors. The numbers correspond to the numbered steps in the main text. DFT = discrete Fourier transform. (This figure is adapted from Morrison et al., 2020.)

3.4 x-vector extraction

3.4.1 Overview

With one feature vector extracted every 10 ms, i.e., 100 feature vectors extracted per second, and recordings potentially including from several seconds to several minutes of speech of the speaker of interest, the number of feature vectors extracted per recording can range from several hundred to tens of thousands. x-vector extraction converts the feature vectors from a recording into a single x-vector. An x-vector has the same length irrespective of the duration of the speech of the speaker of interest on the recording. In addition, x-vector extraction is designed so that, in the multidimensional space of the x-vectors, x-vectors extracted from different recordings of the same speaker will be close to each other whereas x-vectors extracted from recordings of different speakers will be far from each other, i.e., x-vectors have small within-speaker variability and large between-speaker variability. When x-vectors are input to subsequent models that calculate likelihood ratios, those models can therefore produce likelihood-ratio values ranging from much smaller than 1 to much larger than 1 (log-likelihood-ratio values ranging from much less than 0 to much more than 0).

x-vectors are extracted using a deep neural network (DNN), which is an artificial neural network that has multiple layers between the input and output layers. The DNN is trained using a large number of recordings from each of a large number of speakers. The speaker-intrinsic and speaker-extrinsic conditions of the recordings should be diverse, and the speakers should also be diverse. That way, the DNN has the opportunity to learn about both within-speaker variability and between-speaker variability. The DNN has one output node for each speaker in the training set. If the DNN were being used to make a decision as to which of the speakers from the training set was speaking on a recording, the output node with the highest activation would correspond to the speaker with the highest posterior probability. In forensic voice comparison, however, the purpose is not to classify the incoming recording as belonging to one of the speakers on which the system was already trained, the purpose is to calculate a likelihood ratio for the comparison of recordings of the questioned speaker and the known speaker, neither of whom was used to train the system. Therefore, rather than using the output layer of the DNN, the activations of the nodes in a pre-final layer of the DNN are instead used as the values of an x-vector. Because the x-vector layer of the DNN is prior to the output layer, rather then capturing information about the training speakers in particular, it is more abstract and captures information about properties that can distinguish speakers from one another in general.

Key for successful training of a DNN x-vector extractor is to use large amounts of training data. The data should consist of tens of recordings in diverse speaker-intrinsic and speaker-extrinsic conditions from each of thousands of diverse speakers. The data used to train the DNN are not intended to represent the particular population or reflect the particular conditions of the case under consideration. Once it has been trained, however, a DNN can be used to extract x-vectors from recordings that do represent and reflect the populations and conditions specific to a case.

3.4.2 Time-delay DNNs

Compared to the current state of the art, the architecture of DNNs initially used for x-vector extraction was relatively simple. Figure 3 provides a simplified schematic of the architecture of a DNN based on the description in Snyder et al. (2017). The squares in the bottom row of the figure represent feature vectors. For the “frame level” of the DNN, only the time dimension of the feature vectors is shown. Each layer of the frame level includes a parallel set of nodes and connections for each step in the frequency dimension of the feature vectors. The second row from the bottom of the figure represents the input layer of the DNN. The circles represent nodes. Each node is connected to multiple feature vectors from adjacent time steps. The “activation” of a node (the value that is passed to the next layer of the DNN), is a function of the weighted sum of the input values to that node (the function used is often non linear, and different functions may be used for different layers in the DNN). The weights can be thought of as properties of the connections feeding into the node of interest from nodes in the previous layer (or, for the input layer, from the feature matrix) – a higher weight is associated with a stronger connection (the analogy is with the synapses between biological neurons). Progressing up through the frame level of the DNN, each layer combines information from nodes corresponding to multiple time steps in the preceding layer until the time dimension is collapsed to a single node in the time dimension (not shown in Figure 3, there is still a node for each step in the frequency dimension).

<Figure 3 about here>

Figure 3. Simplified schematic of the architecture of a time-delay DNN used for x-vector extraction. (This figure is adapted from Morrison et al., 2020.)

The frame level of the DNN combines information from a total of 15 time steps, 150 ms if feature vectors are extracted every 10 ms. Recordings of speakers of interest are longer than 150 ms. Advancing one time step at a time, all the feature vectors from the speaker of interest on a recording are sequentially presented as input to the DNN, and the “statistics-pooling layer” calculates the mean and standard-deviation values of the activations of the nodes in the immediately preceding layer. There is one mean node and one standard-deviation node for each frequency-step node in the immediately preceding layer. The circles in the “segmental level” of the figure represent additional layers of nodes that process information output by the statistics-pooling layer. This is a fully-connected feed-forward network (within the network, every node in a layer is connected to every node in the immediately preceding layer and every node in the immediately following layer). Prior to the output layer, there is a bottleneck layer (it has fewer nodes than the preceding layer or the following layer), and the activations of the nodes in this layer are used as the values of an x-vector.

To train the DNN, the weights are randomly initialized, a recording is presented, and the weights are adjusted to increase the relative level of activation of the output node corresponding to the speaker on the recording. This is repeated multiple times with tens of recordings from each of thousands of speakers.

To extract an x-vector, a recording is presented to the DNN, and the resulting activations of the nodes of the “x-vector layer” are used as the values of the x-vector. The output layer of the DNN is not used.

3.4.3 Residual networks (ResNets)

Current state-of-the-art x-vector extraction uses more complex DNNs called Residual Networks (ResNets; He et al., 2016). The example system uses a variant of the ResNet34 architecture described in Chung et al. (2020a, 2020b). The details of sizes of layers etc. in the following paragraphs are those of the example system.

Each input-layer node of the ResNet receives input from a square “patch” of feature values which covers 7 time steps by 7 frequency steps in the feature matrix, see Figure 4. There is one input-layer column for each time step in the feature matrix and one input-layer row for every other frequency step in the feature matrix – the “stride” is 1 in the time dimension and 2 in the feature dimension. The length of the input-layer rows, T, is 400. The length of the input-layer columns, F, is 20 (the length of each feature vector is 40). A node in the input layer nominally corresponds to the feature-matrix time and frequency step that is in the center of the 7×7 patch. If the center of a patch is near the edge of the feature matrix (in time or frequency steps), the part of the patch that extends beyond the edge of the feature matrix feeds in values of 0 (the feature matrix is padded with zeros). The set of connection weights between each node in the input layer and its corresponding patch of feature values is called a “kernel”. The same kernel is used for all input-layer nodes (the kernel is convolved with the matrix of feature values). Additional kernels are created by initializing the connections with different sets of weights. Each additional kernel is used for all nodes in an additional input layer that is parallel to the first input layer. Each parallel input layer creates a “channel” which is parallel to the other channels. The number of input channels, C, is 16.

<Figure 4 about here>

Figure 4. Simplified schematic of the feature vectors and the input layer of a ResNet

DNN used for x-vector extraction. Only one channel is shown. (This figure is reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

Figure 5 provides a simplified schematic of the architecture of the ResNet used for x-vector extraction by the example system. The ResNet consists of a series of “groups”, each group consists of a series of “blocks”, and each block consists of a series of layers.

<Figure 5 about here>

Figure 5. Simplified schematic of the architecture of a ResNet DNN used for x-vector extraction. (This figure is reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

Figure 6 provides a simplified schematic of the architecture of a block. The first and second layer of each block are similar to the input layer of the ResNet in that each node receives input from a patch of nodes in the immediately preceding layer. These patches cover 3 time steps by 3 frequency steps by the full number of channels ($3 \times 3 \times C$). In Groups 2 and 3, the stride for the first layer of the first block is 2 for both the time and frequency dimensions (hence the size of both these dimensions is halved). For the second layer of the first block in each of Group 2 and 3, and for both the first and second layers of all other blocks in all groups, the stride is 1 in each dimension (hence the size of both these dimensions is unchanged). For the first layer of each of Groups 2 through 4, two kernels are applied to the output of the previous group. This results in a doubling of the number of channels. The sizes of dimensions T and F and the number of channels C within each group are provided in Table 1.

<Figure 6 about here>

Figure 6. Simplified schematic of the architecture of a block within a ResNet DNN

used for x-vector extraction. Only one channel is shown. The "input" is the last layer of the previous block. (This figure is reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

Table 1. Sizes of the dimensions of the components and subcomponents of the ResNet DNN used by the example system for x-vector extraction.

<Table 1 about here>

After its first two layers, each block has a one-dimensional "squeeze-excitation network". Each node in the input layer of this network calculates the mean value of all the nodes in the previous layer belonging to a single channel, e.g., in the first block there are 16 channels therefore there are 16 nodes in the input layer of the squeeze-excitation network. The network then has a bottleneck layer and an output layer. The output layer has the same number of nodes as the input layer, i.e., one per channel. The activations of the nodes in the output layer are used to weight the channels relative to one another. This focuses "attention" on the channels that are more useful for distinguishing speakers from one another (Cai et al., 2018).

For each channel, the output of a block is the elementwise summation of the channel-weighted output of the block's second layer and of the original input to the block. If there is a difference in the number of time or frequency steps or the number of channels between the previous block and the current block, in order to be able to perform the elementwise summation, the input to the block is processed through a set of kernels that alter its dimensions to match those of the current block. This set of kernels is independent of other sets of kernels. The addition of the input to a block to what would otherwise be its output is the "residual" that give ResNets their name.

Figure 7 provides a simplified schematic of the final stages of the ResNet, which consist of a “statistics-pooling block”, an “x-vector layer”, and an “output layer”. For each channel, the first layer of the statistic-pooling block collapses the frequency dimension by calculating the mean of the column of frequency values corresponding to each time step of the immediately preceding layer. This results in a two-dimensional $T \times C$ layer of nodes.

After the first layer of the statistic-pooling block, similar to the squeeze-excitation networks in earlier blocks, there is a one-dimensional channel-attention layer which is the same length as the number of channels. Unlike the squeeze-excitation networks, the channel-attention layer is a single layer and is only connected to a higher layer, it does not have input from a lower layer. The activations of the nodes in the channel-attention layer are therefore learned during training, and thereafter are fixed. The activations of the nodes in the channel-attention layer are used to weight the channels of the statistic-pooling block’s first layer. The result, the second layer, is a one-dimensional layer that is the same length as the number of time steps, and in which the activation of each node is a function applied to the weighted sum of the activations of the first layer’s nodes at the corresponding time step. The activations of the nodes in the second layer are then used to weight the time steps of the statistic-pooling block’s first layer. The result, the third layer, is a one-dimensional layer that is the same length as the number of channels, and in which the activation of each node is the weighted sum of the activations of the first layer’s nodes for the corresponding channel. The third layer is the output layer of the statistics-pooling block. It has combined information from across both time and frequency.

The output layer of the statistics-pooling block is fully connected to the x-vector layer. The x-vector layer of the example system has 512 nodes. The x-vector layer is fully connected to the ResNet’s output layer. The output layer has one node for each speaker in the training data.

<Figure 7 about here>

Figure 7. Simplified schematic of the final stages of a ResNet DNN used for x-vector extraction. The final stages include the pooling block, the x-vector layer, and the output layer. Multiple channels are shown. The “input” are the last layers of the previous block from each of the multiple channels. “×” indicates matrix multiplication. (This figure is reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

For extraction of x-vectors, recordings longer (or shorter) than 400 feature vectors can be presented to the ResNet. Since the same kernels are used for each node in the input layer, the number of nodes in the rows of the input layer can be increased (or decreased) to accommodate the number of feature vectors, and the kernels simply repeated for each node – no retraining is needed to accommodate the different number of feature vectors. The number of time steps in higher layers can likewise be increased (or decreased) without the need for retraining. The statistics-pooling block collapses the time dimension (and the frequency dimension) so that the final one-dimensional layers of the ResNet have the same numbers of nodes irrespective of the number of feature vectors and the number of time steps in earlier layers.

3.5 LDA

From Stage 6 onward, the data used for training or adapting the backend models should be representative of the relevant population for the case and should reflect the conditions of the questioned-speaker and known-speaker recordings for the case, including any mismatch in conditions between the questioned-speaker and known-speaker recordings.

Linear discriminant functions (see Klecka, 1980) are used for mismatch-compensation and to reduce the number of dimensions of the x-vector. This process is referred to in the automatic-speaker-recognition literature as LDA. The example system uses the algorithm described in Hastie et al. (2009, §4.3), and the x-vectors are reduced from 512 to 120 dimensions.

The number of recordings available for training that represent the relevant population for a case and that reflect the conditions of the questioned-speaker and known-speaker recordings for a case is usually relatively low. In addition to using x-vectors from recordings that actually reflect the population and conditions for the case (in-domain data), x-vectors from a large number of non-case-specific recordings from a large number of speakers (out-of-domain data) can be adapted to simulate this population and these conditions. The correlation alignment (CORAL) algorithm (Sun et al., 2017; Alam et al., 2018) linearly shifts and scales the out-of-domain data so that their total covariance matrix (within-speaker plus between-speaker covariance matrix) matches that estimated from the in-domain data. The example system uses the CORAL algorithm described in Alam et al. (2018). The original in-domain data plus the adapted data are then used to calculate the linear discriminant functions. An alternative method would be to use the CORAL+ algorithm to adapt the PLDA model (Lee et al., 2019).

The in-domain and adapted x-vectors of the training data are transformed using the linear discriminant functions. The x-vectors that will be used for calibration and validation are transformed using the same linear discriminant functions.

3.6 PLDA

The post-LDA in-domain and adapted x-vectors are used to train a model that in the automatic-speaker-recognition literature is known as PLDA (Prince & Elder, 2007; Kenny, 2010; Brümmer & de Villiers, 2010; Sizov et al., 2014). The example system implements the two-covariance variant of PLDA described in Brümmer & de Villiers (2010). This is the same as the common-source likelihood-ratio model described in Ommen & Saunders (2021) and the multivariate normal procedure described in Aitken & Lucy (2014). For ease of explanation, a univariate version of the two-covariance PLDA is described below. The analogous multivariate version is then presented.

Prior to training the PLDA model, to better fit the assumptions of the model, the post-LDA x-vectors in the training data are centered, “whitened” (i.e., rotated and scaled so that for the entire training set the variance in each dimension is 1 and the covariance between dimensions is 0, note that these are transformations based on the between-plus-within-source covariance matrix, not either of the within-source or between-source covariance matrices alone), then scaled to unit length in the Euclidian multidimensional space (García-Romero & Espy-Wilson, 2011). The x-vectors that will be used for calibration and validation are transformed using the centering and whitening functions derived from the training data, and then scaled to unit length.

In general, a common-source likelihood-ratio model has the form given in Equation (1) above. The two-covariance PLDA model assumes Gaussian distributions for same-speaker and different-speaker models M_s and M_d, and assumes that all speakers have the same within-speaker variance, see Equation (2), in which λ is an uncalibrated likelihood-ratio value, $f(x|\mu,\sigma^2)$ is a Gaussian probability-density function (parametrized using mean and variance), x_q and x_k are the questioned-speaker and known-speaker (post-LDA post-centering-whitening-and-scaling) x-vectors respectively, μ_r is the relevant-population mean, μ_i and μ_j are means for arbitrary individual speakers, and σ_w^2 and σ_b^2 are the within-speaker variance and the between-speaker variance respectively. σ_w^2, and σ_b^2 are estimated using the training data. σ_w^2 is estimated as the pooled-within-speaker variance. Since the training data are centered, $\mu_\mathrm{r} = 0$.

(2)

$$\lambda = \frac{\int f\left(x_\mathrm{q}\middle|\mu_i,\sigma_\mathrm{w}^2\right) f\left(x_\mathrm{k}\middle|\mu_i,\sigma_\mathrm{w}^2\right) f\left(\mu_i\middle|\mu_\mathrm{r},\sigma_\mathrm{b}^2\right) d\mu_i}{\int f\left(x_\mathrm{q}\middle|\mu_i,\sigma_\mathrm{w}^2\right) f\left(\mu_i\middle|\mu_\mathrm{r},\sigma_\mathrm{b}^2\right) d\mu_i \int f\left(x_\mathrm{k}\middle|\mu_j,\sigma_\mathrm{w}^2\right) f\left(\mu_j\middle|\mu_\mathrm{r},\sigma_\mathrm{b}^2\right) d\mu_j}$$

$$= \frac{f\left(\begin{bmatrix} x_\mathrm{q} \\ x_\mathrm{k} \end{bmatrix}\middle|\begin{bmatrix} \mu_\mathrm{r} \\ \mu_\mathrm{r} \end{bmatrix}, \begin{bmatrix} \sigma_\mathrm{w}^2+\sigma_\mathrm{b}^2 & \sigma_\mathrm{b}^2 \\ \sigma_\mathrm{b}^2 & \sigma_\mathrm{w}^2+\sigma_\mathrm{b}^2 \end{bmatrix}\right)}{f\left(x_\mathrm{q}\middle|\mu_\mathrm{r},\sigma_\mathrm{w}^2+\sigma_\mathrm{b}^2\right) f\left(x_\mathrm{k}\middle|\mu_\mathrm{r},\sigma_\mathrm{w}^2+\sigma_\mathrm{b}^2\right)}$$

$$= \frac{f\left(\begin{bmatrix} x_\mathrm{q} \\ x_\mathrm{k} \end{bmatrix} \middle| \begin{bmatrix} \mu_\mathrm{r} \\ \mu_\mathrm{r} \end{bmatrix}, \begin{bmatrix} \sigma_\mathrm{w}^2 + \sigma_\mathrm{b}^2 & \sigma_\mathrm{b}^2 \\ \sigma_\mathrm{b}^2 & \sigma_\mathrm{w}^2 + \sigma_\mathrm{b}^2 \end{bmatrix}\right)}{f\left(\begin{bmatrix} x_\mathrm{q} \\ x_\mathrm{k} \end{bmatrix} \middle| \begin{bmatrix} \mu_\mathrm{r} \\ \mu_\mathrm{r} \end{bmatrix}, \begin{bmatrix} \sigma_\mathrm{w}^2 + \sigma_\mathrm{b}^2 & 0 \\ 0 & \sigma_\mathrm{w}^2 + \sigma_\mathrm{b}^2 \end{bmatrix}\right)}$$

The numerator of Equation (2) integrates over all possible values for individual-speaker means given the between-speaker distribution, with the constraint that x_q and x_k come from the same speaker. The denominator of Equation (2) integrates over all possible values for individual-speaker means given the between-speaker distribution, but does so independently for x_q and for x_k. The solutions to the integrals can be expressed as bivariate Gaussian distributions in which for the same-speaker model (the numerator model) the covariances equal the between-speaker variance, σ_b^2, but for the different-speaker model (the denominator model) the covariances are zero. This reflects the logic that the values of x-vectors from different recordings of the same speaker are expected to be correlated, but the values of x-vectors from recordings of different speakers are not expected to be correlated. This is graphically represented in Figure 8, in which the different-speaker model has round contours, but the same-speaker model has elliptical contours with their major axes in the direction of the positively correlated diagonal.

<Figure 8 about here>

Figure 8. Graphical representation of the calculation of a likelihood ratio using a univariate two-covariance PLDA model. (This figure is adapted from Morrison et al., 2020.)

The multivariate version of the two-covariance PLDA model is provided in Equation (3), in which $f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian probability-density function, and the scalar features, means, and variances of Equation (2) are replaced by their analogous feature vectors, mean vectors, and covariance matrices.

(3)

$$\lambda = \frac{f\left(\begin{bmatrix}\boldsymbol{x}_\mathrm{q}\\ \boldsymbol{x}_\mathrm{k}\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{\mu}_\mathrm{r}\\ \boldsymbol{\mu}_\mathrm{r}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Sigma}_\mathrm{w}+\boldsymbol{\Sigma}_\mathrm{b} & \boldsymbol{\Sigma}_\mathrm{b}\\ \boldsymbol{\Sigma}_\mathrm{b} & \boldsymbol{\Sigma}_\mathrm{w}+\boldsymbol{\Sigma}_\mathrm{b}\end{bmatrix}\right)}{f\left(\boldsymbol{x}_\mathrm{q}\middle|\boldsymbol{\mu}_\mathrm{r},\boldsymbol{\Sigma}_\mathrm{w}+\boldsymbol{\Sigma}_\mathrm{b}\right)f\left(\boldsymbol{x}_\mathrm{k}\middle|\boldsymbol{\mu}_\mathrm{r},\boldsymbol{\Sigma}_\mathrm{w}+\boldsymbol{\Sigma}_\mathrm{b}\right)}$$

3.7 Calibration

The output of the two-covariance PLDA model was described as an uncalibrated likelihood ratio. This is because the model requires estimation of a large number of parameter values using a limited amount of data. For the example system, it requires the estimation of two covariance matrices in a 120 dimension space, i.e., a total of 14,520 parameter values. The output of the PLDA is therefore not expected to be well calibrated. In the automatic-speaker-recognition literature, the logarithms of the uncalibrated likelihood ratios are called scores, but note that they are not similarity scores (Morrison & Enzinger, 2018; Neumann & Ausdemore, 2020; Neumann et al., 2020).

A calibration model is trained using a dataset which will be called a calibration set. This is a dataset that was not used to train earlier stages of the system and that should be representative of the relevant population for the case and should reflect the conditions of the questioned-speaker and known-speaker recordings for the case. Each speaker in the calibration set should have at least one recording reflecting the conditions of the questioned-speaker recording and at least one recording reflecting the conditions of the known-speaker recording. From the calibration set, pairs of recordings are constructed such that one member of each pair reflects the questioned-speaker conditions and the other member reflects the known-speaker conditions. A large number of pairs should be constructed in which the two recordings in each pair come from the same speaker, and a large number of pairs should be constructed in which the two recordings in each pair come from different speakers. These recordings are processed through Stages 1–5 of our example system, resulting in a set of scores from same-speaker pairs and a set of scores from different-speaker pairs. These scores are univariate, and are used to train a parsimonious calibration model. For the parsimonious univariate model, the number of parameter values to be estimated is small compared to the amount of training data, hence the output of the model is well calibrated. A commonly used model is logistic regression (Pigeon et al., 2000; González-Rodríguez et al., 2007; Morrison, 2013, 2021). Logistic regression fits a linear model in the log-likelihood-ratio space, which only requires the estimation of two parameter values, an intercept β_0 and a slope β_1, see Equation (4).

(4)

$$\log(\Lambda) = \beta_0 + \beta_1 \log(\lambda)$$

The example system uses regularized logistic regression as described in Morrison & Poh (2018). (This model is fitted using a regularized version of the conjugate-gradient method.) Regularization reduces the slope, β_1, with the result that the calibrated log likelihood ratio is closer to the neutral value of 0 than would otherwise be the case (the likelihood-ratio value is closer to 1). This reduces the probability of overstating the strength of evidence in either direction. For the validations conducted using the example system, the regularization weight was set to be equivalent to 1 pseudo-speaker (see Morrison & Poh, 2018, for details).

An additional step that some systems use before calibration is score normalization. Adaptive Symmetric Norm (AS-Norm; Cumani et al., 2011) is currently the standard method for score normalization. Score normalization was not used for the validations conducted using the example system.

4 Validation

4.1 Data and training

The performance of the example system was validated on a benchmark dataset (*forensic_eval_01*) that reflects the conditions of a forensic case. The benchmark dataset and validation protocols are described in Morrison & Enzinger (2016a). The speakers are adult male Australian-English speakers. The questioned-speaker condition reflects a 46 s long landline-telephone call, with background babble noise, saved using lossy compression. The known-speaker condition reflects a 126 s long interview recorded in a reverberant room, with background ventilation-system noise. The durations just stated are for the amount of speech of the speaker of interest after semi-automatic diarization but before applying VAD. The questioned-speaker-condition and known-speaker-condition recordings were recorded on different occasions separated by approximately a week or more. Each speaker in the test set was recorded on at least two occasions. The test set consists of a total of 223 recordings from 61 speakers, 61 in questioned-speaker condition and 162 in known-speaker condition, allowing for the construction of 111 same-speaker pairs of recordings and 6720 different-speaker pairs of recordings (from 3660 pairs of speakers). The dataset also includes a training set consisting of a total of 423 recordings from 105 speakers (191 recordings in questioned-speaker condition and 232 in known-speaker condition).

The x-vector extractor was trained using approximately 1 million recordings total from approximately 6 thousand speakers from the VoxCeleb2 database (Chung et al., 2018; Nagrani et al., 2020). As out-of-domain data for CORAL, approximately 30 thousand recordings total were used from approximately 2.7 thousand speakers from the SRE2018 Test dataset (Greenberg et al., 2020). As in-domain data for LDA and PLDA training, the *forensic_eval_01* training set was used.

For training the calibration model and for validation, the *forensic_eval_01* test set was used. To avoid training and testing on the same data (and in accordance with the recommendations in the “Consensus on validation of forensic voice comparison”; Morrison et al., 2021), leave-one-speaker-out / leave-two-speakers-out cross-validation was used: In a cross-validation loop in which the score to be calibrated was a same-speaker score, e.g., a recording of speaker *A* compared to another recording of speaker

A, all scores that resulted from comparisons in which one or both members of the pair was a recording of speaker *A* were excluded from the data used to train the calibration model (leave-one-speaker-out). In a cross-validation loop in which the score to be calibrated was a different-speaker score, e.g., a recording of speaker *A* compared to a recording of speaker *B*, all scores that resulted from comparisons in which one or both members of the pair was a recording of speaker *A* or a recording of speaker *B* were excluded from the data used to train the calibration model (leave-two-speakers-out).

Prior to use, if not already in this format, all recordings were converted to 8 kHz sampling rate 16 bit quantization PCM.

4.2 Results

A Tippett plot showing validation results is presented in Figure 9. The same-speaker and different-speaker curves have relatively shallow slopes, indicating good performance, and they cross near a log-likelihood-ratio value of 0, indicating good calibration. The Tippett plot indicates that the validation results would support likelihood-ratio values into the thousands in favor of the same-speaker hypothesis and into the tens of thousands in favor of the different-speaker hypothesis ($\log_{10}$ likelihood ratios beyond +3 and −4 respectively).

<Figure 9 about here>

Figure 9. Tippett plot of the results of validating the example system ($E^3FS^3\alpha$) on the *forensic_eval_01* dataset. (This figure is adapted from Weber et al., 2022b.)

A virtual special issue of the journal *Speech Communication*, reports on the validation of several systems using the *forensic_eval_01* dataset. A summary of results is presented in Morrison & Enzinger (2019). Table 2 presents an extract of the C_{llr} results from the virtual special issue, plus the C_{llr} result for the example system. The C_{llr} value for the example system was 0.208. The lower the C_{llr} value, the better the performance of the system. A system that gave no information and always responded with a likelihood ratio of 1 irrespective of the input would have a C_{llr} value of 1. In terms of C_{llr}, the example system performed equally as well as the best-performing system from the virtual special issue, Phonexia SID-BETA4 (Jessen et al., 2019).

Table 2. C_{llr} values from the best-performing version of each system validated in the *Speech Communication* virtual special issue (Morrison & Enzinger, 2019), plus the C_{llr} result for the example system ($E^3FS^3\alpha$).

<Table 2 about here>

4.3 Discussion

The validation results could be used to decide whether the example system should be used to calculate and submit to court a likelihood ratio for comparison of a questioned-speaker recording and a known-speaker recording in a case for which the *forensic_eval_01* dataset represented the relevant population and reflected the conditions for that case. For cases involving other populations and conditions, validations with data representing those populations and reflecting those conditions would need to be conducted before deciding whether the system could be used and the results submitted to courts.

Since the publication of the virtual special issue, improvements may have been made to the actively-developed systems included in Table 2 (Nuance, Phonexia, and VOCALISE), and it may be that the newer versions of these systems would obtain better results.

5 Conclusion

The human-supervised-automatic analytical approach to forensic voice comparison in conjunction with the likelihood-ratio interpretive framework has been described. The description included practitioner tasks, including adoption of the relevant hypotheses for the case, the assessment of the conditions of the questioned-speaker and known-speaker recordings in the case, and the selection of data representing the relevant population and reflecting the conditions for the case. It also included an example forensic-voice-comparison system based on state-of-the-art automatic-speaker-recognition technology, and validation of that system using a benchmark dataset reflecting the conditions of a real forensic case.

6 Relevant webpages

E^3 Forensic Speech Science System (E^3FS^3)
https://e3fs3.forensic-voice-comparison.net/

Virtual special issue of the journal *Speech Communication*: “Multi-laboratory evaluation of forensic voice comparison systems under conditions reflecting those of a real forensic case (forensic_eval_01)”
https://www.sciencedirect.com/journal/speech-communication/special-issue/10KTJHC7HNM

7 References

Aitken, C.G.G. and Lucy, D. (2004). Evaluation of trace evidence in the form of multivariate data. *Applied Statistics* **53**, 109–122. (http://dox.doi.org/10.1046/j.0035-9254.2003.05271.x) [Corrigendum: (2004)

53, 665–666. http://dox.doi.org/10.1111/j.1467-9876.2004.02031.x]

Alam, J., Bhattacharya, G. and Kenny, P. (2018). Speaker verification in mismatched conditions with frustratingly easy domain adaptation. *Proceedings of Odyssey 2018: The speaker and language recognition workshop*, pp. 176–180. (https://doi.org/10.21437/Odyssey.2018-25)

Alam, J., Boulianne, G., Burget, L., Dahmane, M., Díez Sánchez, M., Lozano-Díez, A., Glembek, O., St-Charles, P., Lalonde, M., Matejka P,., Mizera, P., Monteiro, J., Mosner, L., Noiseux, C., Novotný, O., Plchot, O., Rohdin, J., Silnova, A., Slavicek, J., Stafylakis, T., Wang, S. and Zeinali, H. (2020). Analysis of ABC submission to NIST SRE 2019 CMN and VAST challenge. *Proceedings of Odyssey 2020: The speaker and language recognition workshop*, pp. 289–295. (https://doi.org/10.21437/Odyssey.2020-41)

Brümmer, N. and de Villiers, E. (2010). The speaker partitioning problem. *Proceedings of Odyssey 2010: The speaker and language recognition workshop*, pp. 194–201. (https://www.isca-speech.org/archive_open/odyssey_2010/od10_034.html)

Cai, W., Chen, J. and Li, M. (2018). Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. *Proceedings of Odyssey 2018: The speaker and language recognition workshop*, pp. 74–81. (https://10.21437/Odyssey.2018-11)

Chung, J.S., Nagrani, A. and Zisserman, A. (2018). VoxCeleb2: Deep speaker recognition. *Proceedings of Interspeech*, pp. 1086–1090. (https://doi.org/10.21437/Interspeech.2018-1929)

Chung, J.S., Huh, J., Mun, S., Lee, M., Heo, H.S., Choe, S., Ham, C., Jung, S., Lee, B.-J. and Han, I. (2020a). In defence of metric learning for speaker recognition. *Proceedings of Interspeech*, pp. 2977–2981. (https://doi.org/10.21437/Interspeech.2020-1064)

Chung, J.S., Huh, J. and Mun, S. (2020b). Delving into VoxCeleb: Environment invariant speaker recognition. *Proceedings of Odyssey 2020: The speaker and language recognition workshop*, pp. 349–356. (https://doi.org/10.21437/Odyssey.2020-49)

Cumani, A., Batzu, P.D., Colibro, D., Vair, C., Laface, P. and Vasilakakis, V. (2011) Comparison of speaker recognition approaches for real applications. *Proceedings of Interspeech*, pp. 2365–2368. (https://isca-speech.org/archive/interspeech_2011/i11_2365.html)

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **28**, 357–366. (https://doi.org/10.1109/TASSP.1980.1163420)

Diez, M., Burget, L., Wang, S., Rohdin, J. and Černocký H. (2019). Bayesian HMM based x-vector clustering for speaker diarization. *Proceedings of Interspeech*, pp. 346–350. (http://doi.org/10.21437/Interspeech.2019-2813)

Diez, M., Burget, L., Landini, F. and Černocký J. (2020a). Analysis of speaker diarization based on Bayesian HMM with eigenvoice priors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 355–368. (https://doi.org/10.1109/TASLP.2019.2955293)

García-Romero D. and Espy-Wilson C.Y. (2011). Analysis of i-vector length normalization in speaker recognition systems. *Proceedings of Interspeech*, pp. 249–252. (https://doi.org/10.21437/Interspeech.2011-53)

González-Rodríguez, J., Rose P., Ramos, D., Toledano, D.T. and Ortega-García, J. (2007). Emulating DNA: Rigorous quantification of evidential weight in transparent and testable forensic speaker recognition. *IEEE Transactions on Speech and Audio Processing* **15**, 2104–2115. (https://doi.org/10.1109/TASL.2007.902747)

Greenberg, C., Sadjadi, O., Singer, E., Walker, K., Jones, K., Wright, J. and Strassel, S. (2020). 2018 NIST Speaker Recognition Evaluation Test Set (LDC2020S04). Linguistic Data Consortium. (https://catalog.ldc.upenn.edu/LDC2020S04)

Hansen, J.H.L. and Bořil, H. (2018). On the issues of intra-speaker variability and realism in speech, speaker, and language recognition tasks. *Speech Communication* **10**, 94–108. (https://doi.org/10.1016/j.specom.2018.05.004)

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd edn.). New York: Springer.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. (https://doi.org/10.1109/CVPR.2016.90)

Jessen, M., Bortlík, J., Schwarz, P. and Solewicz, Y.A. (2019). Evaluation of Phonexia automatic speaker recognition software under conditions reflecting those of a real forensic voice comparison case (forensic_eval_01). *Speech Communication* **111**, 22–28. (https://doi.org/10.1016/j.specom.2019.05.002)

Kelly, F. and Hansen, H.L. (2021). Analysis and calibration of Lombard effect and whisper for speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing* **29**, 927–942. (https://ieeexplore.ieee.org/document/9330795)

Kenny, P. (2010). Bayesian speaker verification with heavy tailed priors. *Proceedings of Odyssey 2010: The Speaker and Language Recognition Work*shop, paper 14. (https://www.isca-speech.org/archive_open/odyssey_2010/od10_014.html)

Kinnunen, T., Sholokhov, A., el Khoury, E., Thomsen, D.A.L., Sahidullah, M. and Tan, Z-H. (2016). HAPPY team entry to NIST OpenSAD challenge: A fusion of short-term unsupervised and segment i-vector based speech activity detectors. *Proceedings of Interspeech*, pp. 2992–2996. (https://doi.org/10.21437/Interspeech.2016-1281)

Klecka, W.R. (1980). *Discriminant analysis*. Beverly Hills, CA: Sage.

Landini, F., Wang, S., Díez, M., Burget, L., Matejka, P., Zmolíková, K., Mosner, L., Silnova, A., Plchot, O., Novotný, O., Zeinali, H. and Rohdin J. (2020a). BUT system for the second DIHARD speech diarization challenge. *Proceedings of the IEEE International Conference on Digital Signal Processing (ICASSP),* pp. 6529–6533. (https://doi.org/10.1109/ICASSP40776.2020.9054251)

Landini, F., Profant, J., Diez, M. and Burget, L. (2022). Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks. *Computer Speech & Language* **71**, 101254. (https://doi.org/10.1016/j.csl.2021.101254)

Lee, K.A., Wang ,Q. and Koshinaka T. (2019). The CORAL+ algorithm for unsupervised domain adaptation of PLDA. *Proceedings of the IEEE International Conference on Digital Signal Processing (ICASSP),* pp. 5821–5825. (https://doi.org/10.1109/ICASSP.2019.8682852)

Lee, K.A., Yamamoto, H., Okabe, K., Wang, Q., Guo, L., Koshinaka, T., Zhang, J. and Shinoda, K. (2020). NEC-TT system for mixed-bandwidth and multi-domain speaker recognition. *Computer Speech & Language* **61**, 101033. (https://doi.org/10.1016/j.csl.2019.101033)

Matějka, P., Plchot, O., Glembek, O., Burget, L., Rohdin, J., Zeinali, H., Mošner, L., Silnova, A., Novotný, O., Diez, M. and Černocký, J.H. (2020). 13 years of speaker recognition research at BUT, with longitudinal analysis of NIST SRE. *Computer Speech & Language* **63**, 101035. (https://doi.org/10.1016/j.csl.2019.101035)

Morrison, G.S. (2013). Tutorial on logistic-regression calibration and fusion:

converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences* **45**, 173–197. (http://dx.doi.org/10.1080/00450618.2012.733025)

Morrison, G.S. (2021). In the context of forensic casework, are there meaningful metrics of the degree of calibration? *Forensic Science International: Synergy* **3**, 100157. (https://doi.org/10.1016/j.fsisyn.2021.100157)

Morrison, G.S. and Enzinger, E. (2016). Multi-laboratory evaluation of forensic voice comparison systems under conditions reflecting those of a real forensic case (forensic_eval_01) - Introduction. *Speech Communication* **85**, 119–126. (http://dx.doi.org/10.1016/j.specom.2016.07.006)

Morrison, G.S. and Enzinger, E. (2018). Score based procedures for the calculation of forensic likelihood ratios – Scores should take account of both similarity and typicality. *Science & Justice* **58**, 47–58. (http://dx.doi.org/10.1016/j.scijus.2017.06.005)

Morrison, G.S. and Enzinger, E. (2019). Multi-laboratory evaluation of forensic voice comparison systems under conditions reflecting those of a real forensic case (forensic_eval_01) - Conclusion. *Speech Communication* **112**, 37–39. (https://doi.org/10.1016/j.specom.2019.06.007)

Morrison, G.S., Enzinger, E., Hughes, V., Jessen, M., Meuwly, D., Neumann, C., Planting, S., Thompson, W.C., van der Vloed, D., Ypma, R.J.F., Zhang, C., Anonymous, A. and Anonymous, B. (2021). Consensus on validation of forensic voice comparison. *Science & Justice* **61**, 229–309. (https://doi.org/10.1016/j.scijus.2021.02.002)

Morrison, G.S., Enzinger, E., Ramos, D., González-Rodríguez, J. and Lozano-Díez, A. (2020). Statistical models in forensic voice comparison. In Banks, D.L., Kafadar, K., Kaye, D.H. and Tackett, M. (eds.) *Handbook of Forensic Statistics*, pp. 451–497. Boca Raton, FL: CRC. (https://doi.org/10.1201/9780367527709)

Morrison, G.S., Enzinger, E. and Zhang, C. (2016). Refining the relevant population in forensic voice comparison – A response to Hicks et alii (2015) The importance of distinguishing information from evidence/observations when formulating propositions. *Science & Justice* **56**, 492–497. (http://dx.doi.org/10.1016/j.scijus.2016.07.002)

Morrison, G.S. and Kelly, F. (2019). A statistical procedure to adjust for time-interval mismatch in forensic voice comparison. *Speech Communication* **112**, 15–21. (https://doi.org/10.1016/j.specom.2019.07.001)

Morrison, G.S. and Poh, N. (2018). Avoiding overstating the strength of forensic evidence: Shrunk likelihood ratios / Bayes factors. *Science & Justice* **58**, 200–218. (http://dx.doi.org/10.1016/j.scijus.2017.12.005)

Nagrani, A., Chung, J.S., Xie, W. and Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech and Language* **60** 101027. (https://doi.org/10.1016/j.csl.2019.101027)

Nautsch, A., Bamberger, R. and Busch, C. (2016). Decision robustness of voice activity segmentation in unconstrained mobile speaker recognition environments. *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7. (https://doi.org/10.1109/BIOSIG.2016.7736916)

Neumann, C. and Ausdemore, M. (2020). Defence against the modern arts: The curse of statistics – Part II: ‘Score-based likelihood ratios’. *Law, Probability and Risk* **19**, 21–42. (http://dx.doi.org/10.1093/lpr/mgaa006)

Neumann, C., Hendricks, J. and Ausdemore, M. (2020). Statistical support for conclusions in fingerprint examinations. In Banks, D., Kafadar, K., Kaye, D.H., Tackett, M. (eds.) *Handbook of forensic statistics*, pp. 277–324. Boca Raton, FL: CRC. (https://doi.org/10.1201/9780367527709)

Ommen, D.M. and Saunders, C.P. (2021). A problem in forensic science highlighting the differences between the Bayes factor and likelihood ratio. *Statistical Science* **36**, 344–359. (https://doi.org/10.1214/20-STS805)

Pigeon, S., Druyts, P. and Verlinde, P. (2000). Applying logistic regression to the fusion of the NIST'99 1-speaker submissions. *Digital Signal Processing* **10**, 237–248. (https://doi.org/10.1006/dspr.1999.0358)

Prince, S.J.D. and Elder, J.H. (2007). Probabilistic linear discriminant analysis for inferences about identity. *Proceedings of the IEEE 11th International Conference on Computer Vision*, pp. 1–8. (https://doi.org/10.1109/ICCV.2007.4409052)

Sizov, A., Lee, K.A. and Kinnunen, T. (2014). Unifying probabilistic linear discriminant analysis variants in biometric authentication. In Fränti, P., Brown, G., Loog, M., Escolano, F. and Pelillo, M. (eds.) *Structural, syntactic, and statistical pattern recognition*, pp. 464–475. Berlin: Springer. (https://doi.org/10.1007/978-3-662-44415-3_47)

Snyder, D., García-Romero, D., Povey, D. and Khudanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. *Proceedings of Interspeech*, pp. 999–1003. (https://doi.org/10.21437/Interspeech.2017-620)

Sun, B., Feng, J. and Saenko, K. (2017). Correlation alignment for unsupervised domain adaptation. In: Csurka G. (ed.) *Domain adaptation in computer vision applications. Advances in computer vision and pattern recognition*. Cham: Springer. (https://doi.org/10.1007/978-3-319-58347-1_8)

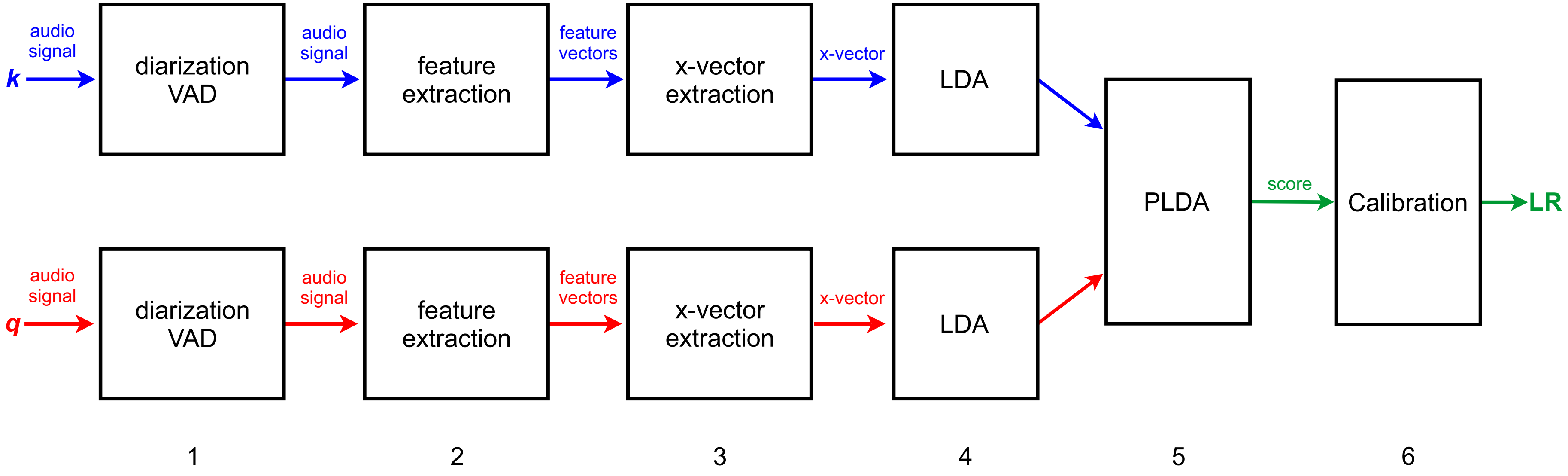
Tan, Z., Sarkar, A.K. and Dehak, N. (2020). rVAD: An unsupervised segment-based robust voice activity detection method. *Computer Speech & Language* **59**, 1–21. (https://doi.org/10.1016/j.csl.2019.06.005)

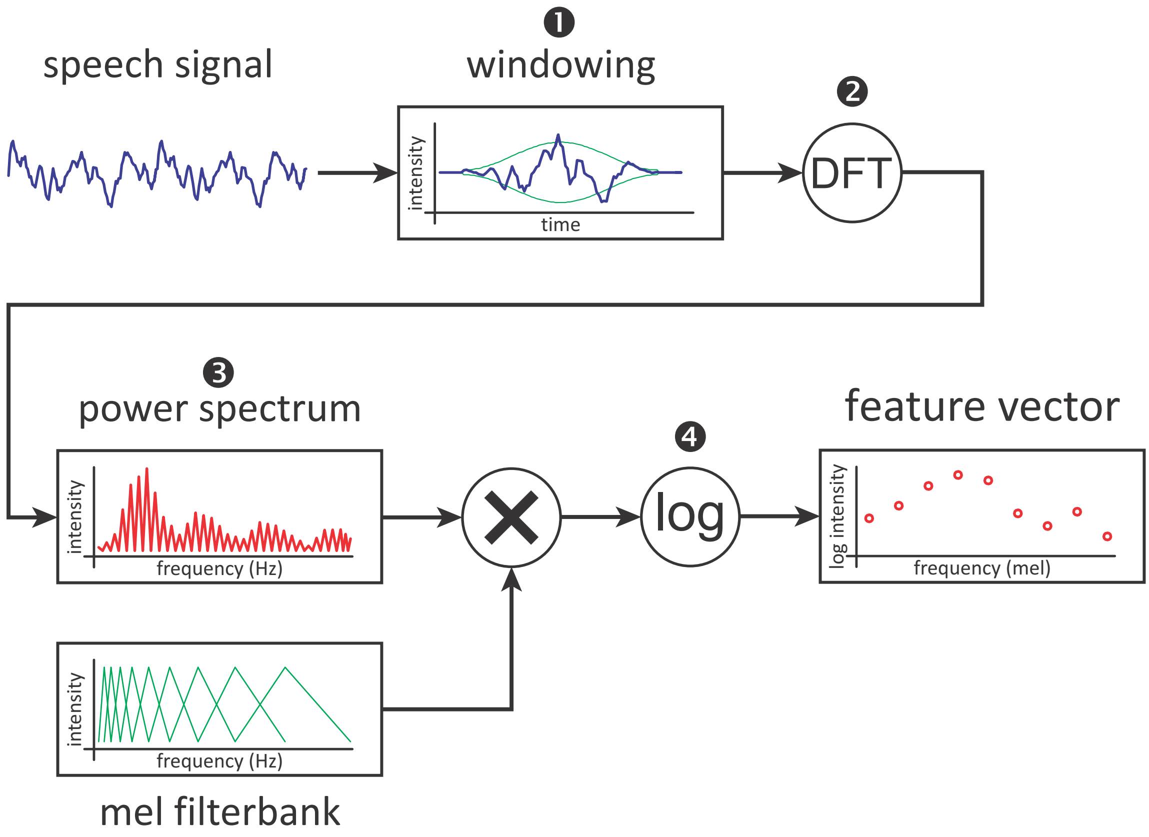
Villalba, J., Chen, N., Snyder, D., García-Romero, D., McCree, A., Sell, G., Borgstrom, J., García-Perera, L.P., Richardson, F., Dehak R., Torres-Carrasquillo, P.A. and Dehak, N. (2020). State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations. *Computer Speech & Language* **60**, 101026. (https://doi.org/10.1016/j.csl.2019.101026)

Weber, P., Enzinger, E., and Morrison, G.S. (2022a) E^3 forensic speech science system (E^3FS^3): Technical report on design and implementation of software tools. (https://forensic-voice-comparison.net/E3FS3/)

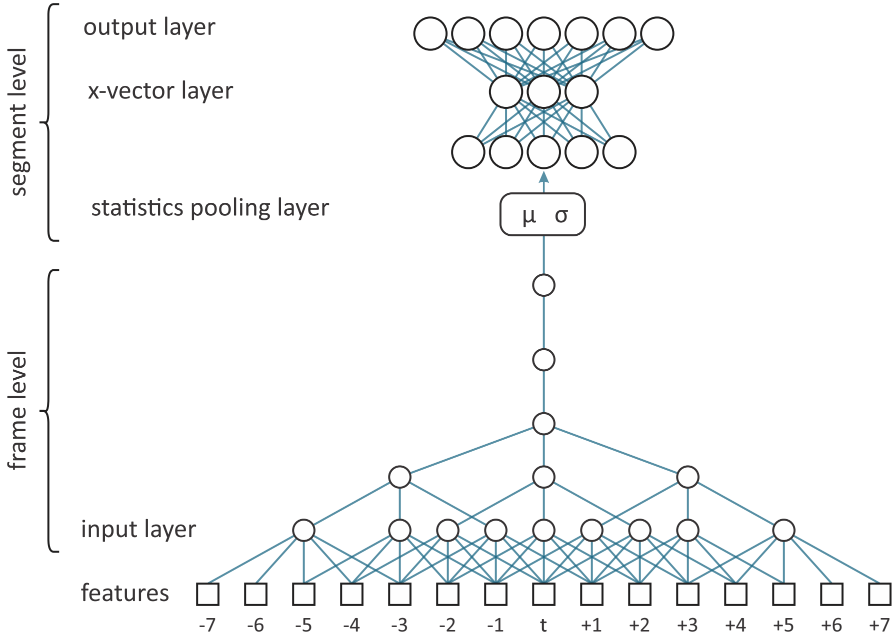
Weber, P., Enzinger, E., Labrador, B., Lozano-Díez, A., Ramos, D., González-Rodríguez, J. and Morrison G.S. (2022b). Validation of the alpha version of the E^3 forensic speech science system (E^3FS^3) core software tools. *Forensic Science International: Synergy* **4**, 100223. (https://doi.org/10.1016/j.fsisyn.2022.100223)

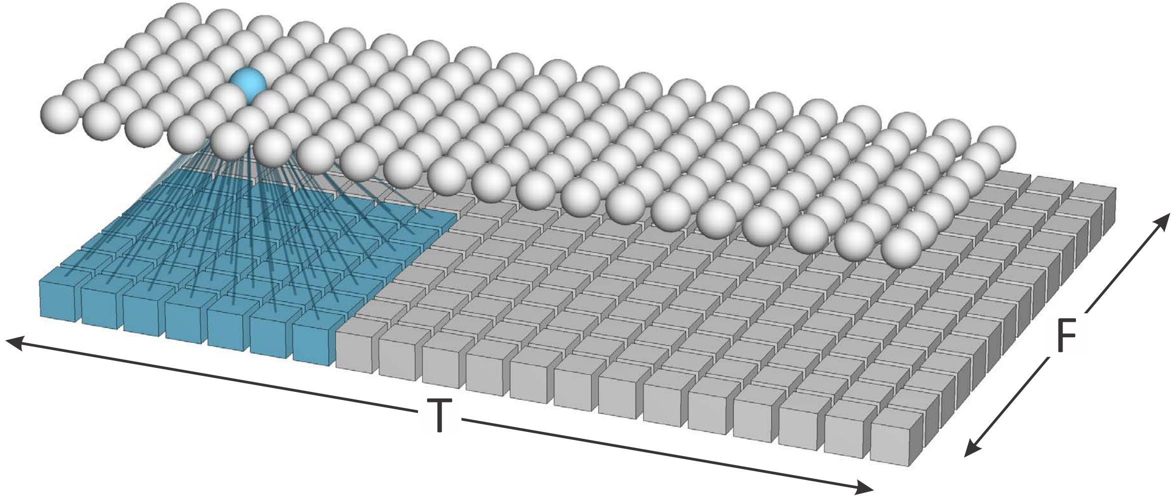
Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. (A.), Moore, G., Odell, J., Ollason, D., Povey, D., Ragni, A., Valtchev, V., Woodland, P. and Zhang, C., (2015). The HTK book. Cambridge University Engineering Department. (https://htk.eng.cam.ac.uk/)

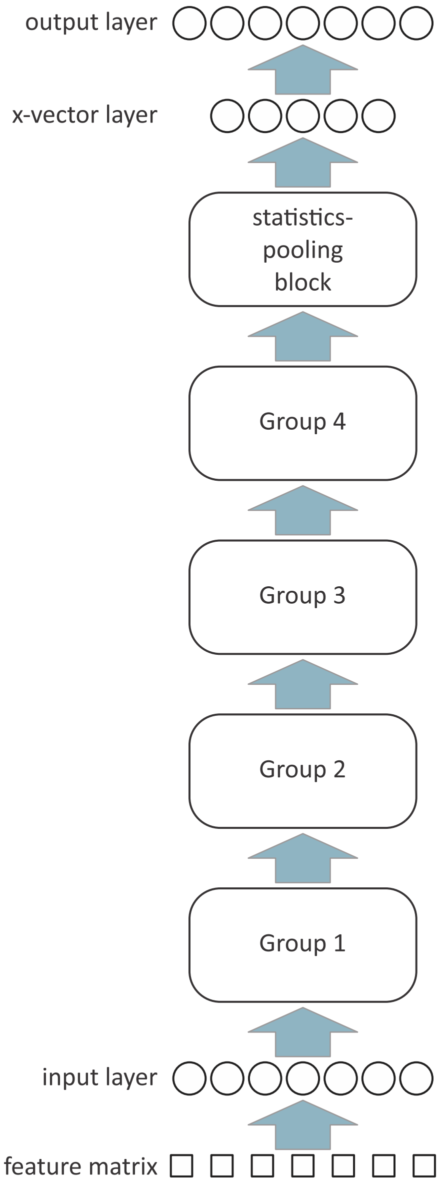

audio signal
k
diarization VAD
audio signal
feature extraction
feature vectors
x-vector extraction
x-vector
LDA
audio signal
q
diarization VAD
audio signal
feature extraction
feature vectors
x-vector extraction
x-vector
LDA
PLDA
score
Calibration
LR
1
2
3
4
5
6

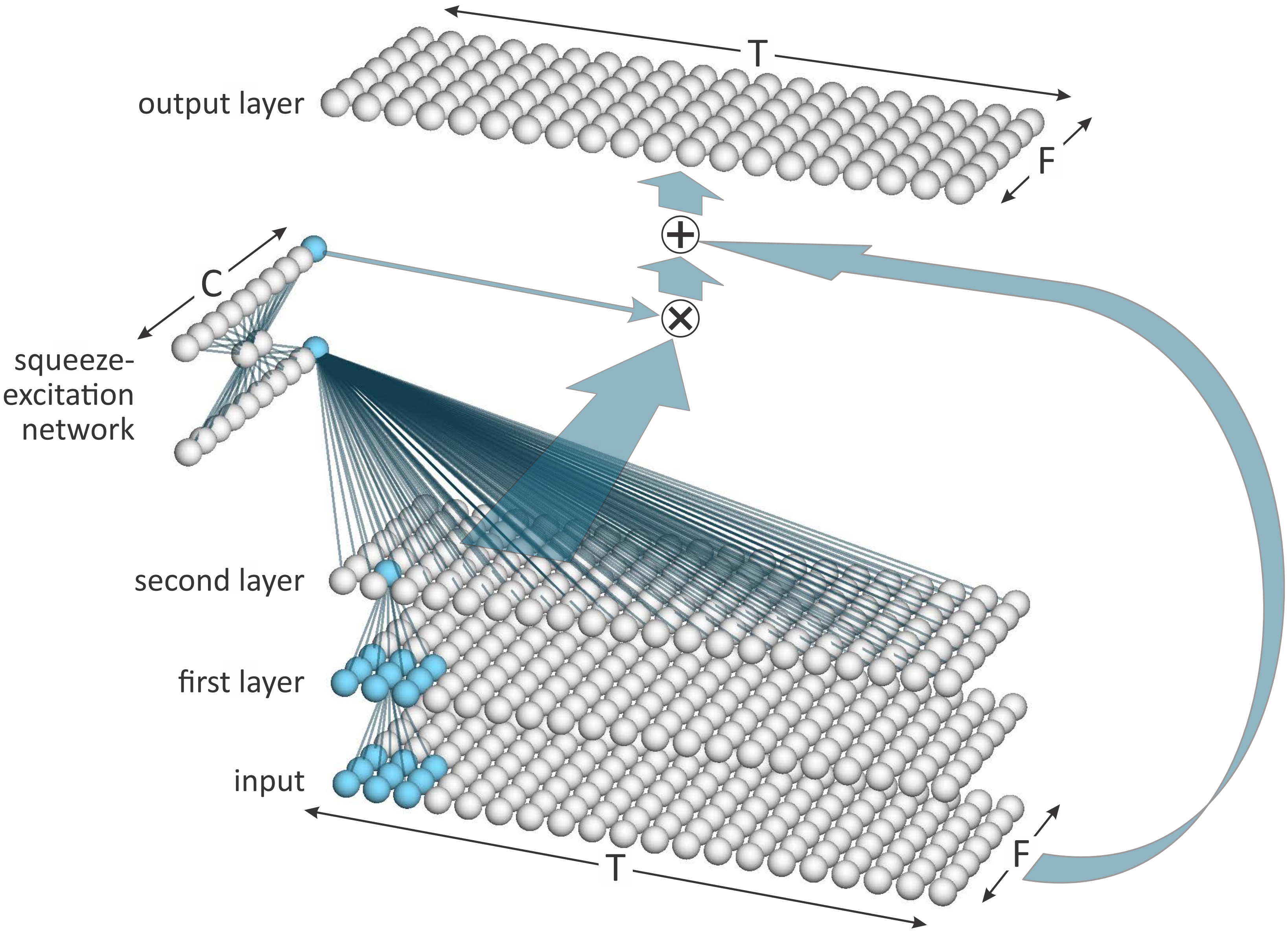


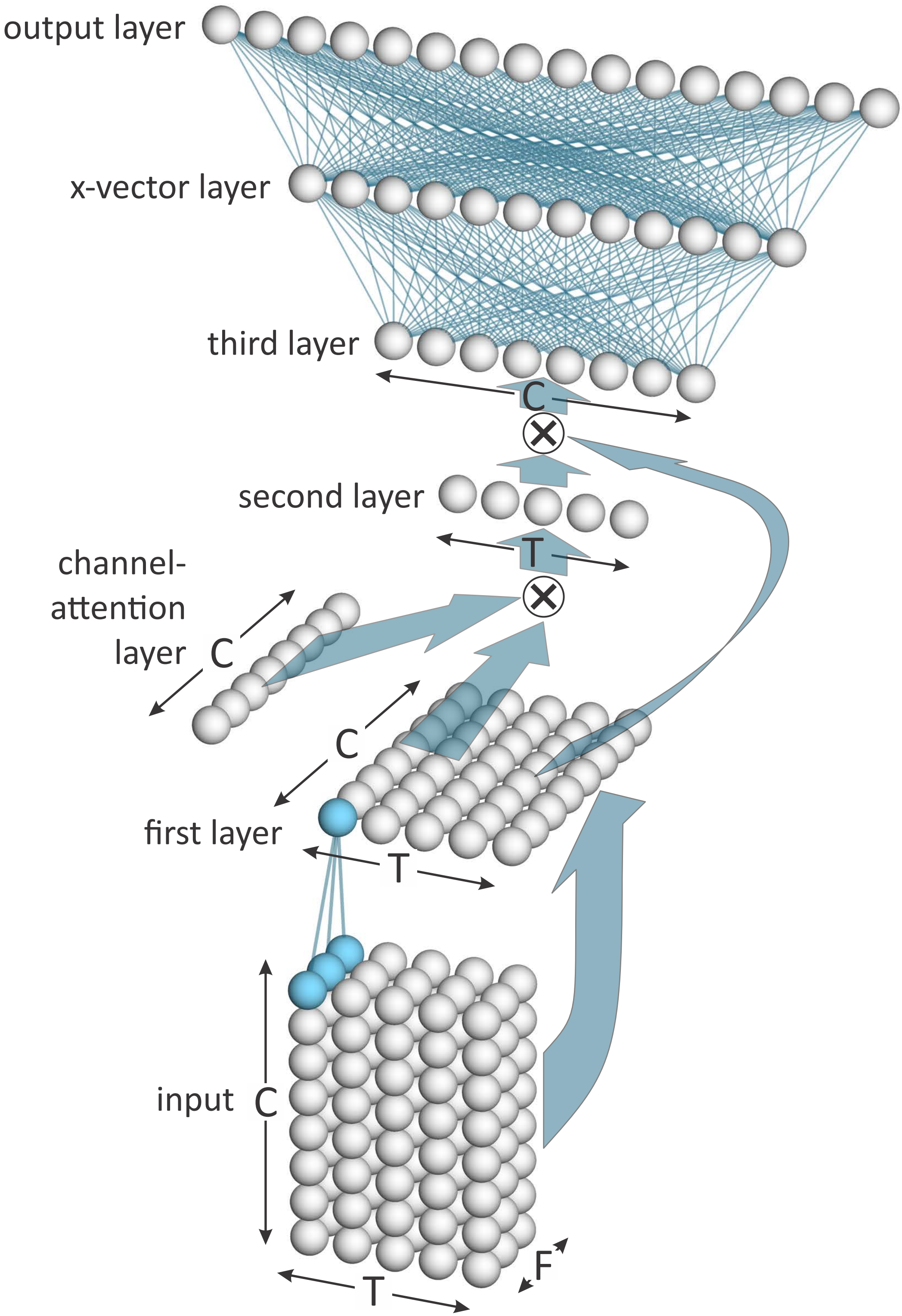
speech signal
❶
windowing
intensity
time
❷
DFT
❸
power spectrum
intensity
frequency (Hz)
❹
log
feature vector
log intensity
frequency (mel)
intensity
frequency (Hz)
mel filterbank

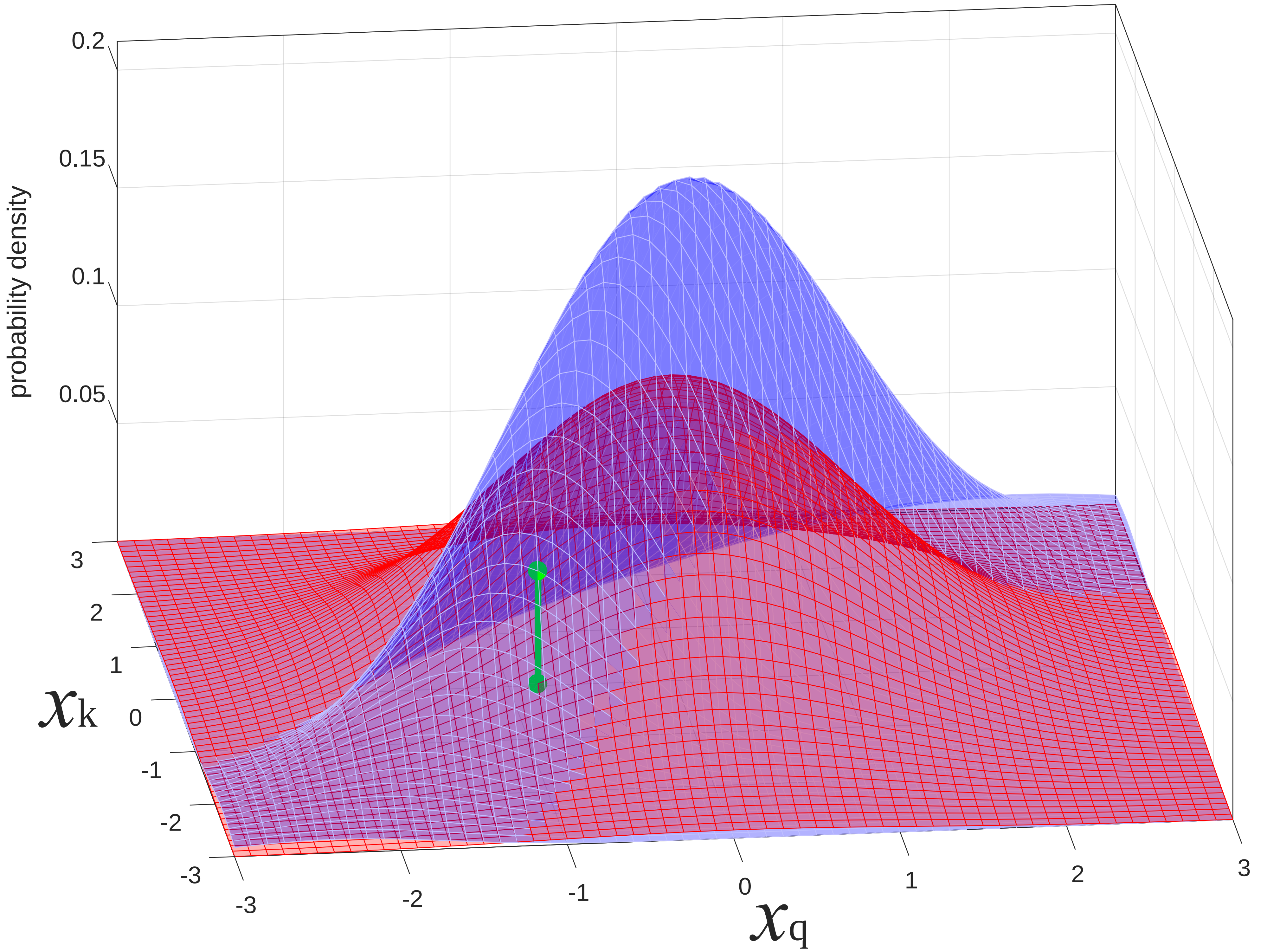

segment level
output layer
x-vector layer
statistics pooling layer
μ σ
frame level
input layer
features
-7
-6
-5
-4
-3
-2
-1
t
+1
+2
+3
+4
+5
+6
+7


T
F

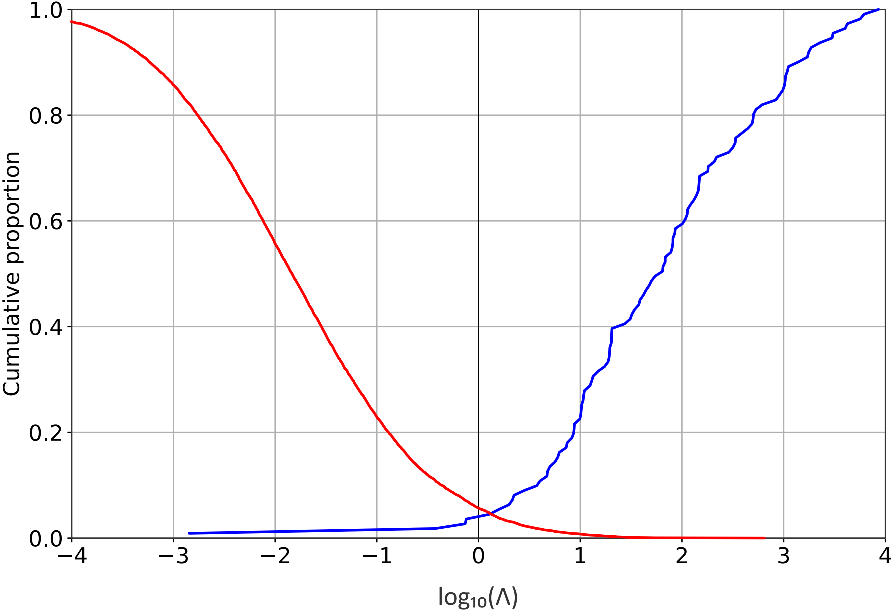

output layer
x-vector layer
statistics-pooling block
Group 4
Group 3
Group 2
Group 1
input layer
feature matrix


T
output layer
F
C
squeeze-
excitation
network
second layer
first layer
input
T
F


output layer
x-vector layer
third layer
C
second layer
T
channel-
attention
layer
C
C
first layer
T
input
C
T
F



probability density
0.2
0.15
0.1
0.05
3
2
1
0
-1
-2
-3
x_k
-3
-2
-1
0
1
2
3
x_q



1.0
0.8
0.6
0.4
0.2
0.0
Cumulative proportion
−4
−3
−2
−1
0
1
2
3
4
log₁₀(Λ)

Table 1. Dimensions of the components of the ResNet DNN used by the example system for x-vector extraction.

Component	**Subcomponent**	**Dimensions**		
		time ***T***	**frequency** ***F***	**channels** ***C***
Feature vectors	–	400	40	1
Input layer	–	400	20	16
Group 1	3 blocks	400	20	16
Group 2	4 blocks	200	10	32
Group 3	6 blocks	100	5	64
Group 4	3 blocks	100	5	128
Statistics-pooling block	Layer 1	100	1	128
	Channel-attention layer	1	1	128
	Layer 2	100	1	1
	Layer 3	1	1	128
x-vector layer	–	1	1	512
Output layer	–	1	1	Number of training speakers

Table 2. C_{llr} values from the best-performing version of each system validated in the *Speech Communication* virtual special issue (Morrison & Enzinger, 2019), plus the C_{llr} result for the example system ($E^3FS^3\alpha$).

System	**Type**	C_{llr}
Batvox 3.1	GMM-UBM	0.593
MSR GMM-UBM	GMM-UBM	0.576
MSR GMM i-vector	GMM i-vector	0.449
Batvox 4.1	GMM i-vector	0.365
Phonexia XL3	DNN bottleneck	0.294
Nuance 9.2	GMM i-vector	0.285
VOCALISE 2017B	GMM i-vector	0.267
Nuance 11.1	DNN senone	0.255
VOCALISE 2019A	x-vector	0.246
$\mathbf{E^3FS^3\alpha}$	x-vector	0.208
Phonexia BETA4	x-vector	0.207