Universitatea ”Politehnica” din București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Caller ID folosind vocea, aplicație pentru Call Center

Lucrare de disertație

prezentată ca cerință pentru obținerea titlului de

Master în domeniul *Inginerie electronică şi telecomunicaţii*

programul de studii de masterat

Tehnologii multimedia în aplicaţii de biometrie şi securitatea informaţiei

Conducători științifici

As. Dr. Ing. Horia CUCU

Ing. Viorel STAN – VivaCredit IFN

Absolvent

Călin NECULA

2014

Tema proiectului

Declaratie de originalitate

Cuprins

Lista figurilor

Lista tabelelor

Lista acronimelor

GMM – model mixturi Gaussiene – *Gaussian Mixture Models*

HMM – model Markov ascuns – *Hidden Markov Models*

CMS – îndepărtarea mediei cepstrale-*Ceptral Mean Substraction*

EM – *Expectation Maximization*

MAP – *Maximimum A-Posteriori*

UBM – modelul universal -*Universal Background Model*

SIP - protocol de initializare a sesiunii -Session Initiation Protocol

JFA - *Joint Factor Analysis*

PCM – *Pulse Code Modulation*

WCCN – normalizarea intra clasă a canalului

LPC - *Linear Predictive Coding*

PLP - *Perceptual Linear Predictive*

ROC – *Relative Operating Caracteristic*

EER – *Equal Error Rate*

FAR - rata de falsă acceptare- *False Acceptance Rate*

FRR – rata de falsă respingere– *False Rejection Rate*

SDP - protocol de descriere a sesiunii- *Session Description Protocol*

RTP - protocol de transport in timp real-*Realtime Transport Protocol*

Capitolul 1

Introducere

1.1. Obiectivul lucrării

Această lucrare de disertație are ca scop proiectarea și realizarea practică a unei aplicații software pentru recunoașterea automată a vorbitorului din cadrul convorbirilor telefonice ale unui callcenter cu scopul de a reduce gradul de fraudă prin detecția persoanelor cu identități false multiple.

Pornind de la această premiză obiectivul lucrării se poate împărți în două obiective principale. Primul obiectiv este găsirea unei metode state-of-the-art de recunoaștere de vorbitor care să funcționeze pe o bază cât mai mare de vorbitori, să poată fi aplicată independent de vorbitor, să fie robustă la zgomot și în același timp să ofere un răspuns cât mai corect într-un timp cât mai scurt. Problema principală este populația mare de clienți ce trebuie să fie antrenată ce aduce după sine alte probleme precum riscul ridicat de impostori sau acceptări false, timpul necesar pentru a identifica un vorbitor dar și timpul necesar antrenării unei persoane noi.

Al doilea obiectiv este studiul metodei de integrare a sistemului de recunoaștere de vorbitor cu centrala telefonică. Problema principală este modul în care clientul comunică cu callcenterul, canalul de voce de la client trebuind să fie "ascultat" de operatorul de callcenter dar și de sistemul de recunoaștere vorbitor.

1.2. Aplicații ale recunoașterii vorbitorului

Recunoașterea vorbitorului are o aplicabilitate foarte mare în special datorită faptului că această metoda este neintruzivă, universal acceptată, cu o acuratețe acceptabilă, iar eșantioanele de vorbire sunt ușor de prelevat, fie prin telefon fie cu ajutorul unui microfon [1].

Sistemele de recunoaștere de vorbitor au început să fie din ce în ce mai mult în aplicații de operațiuni bancare prin telefon. În Mai 2013 s-a anunțat că se poate face recunoașterea vorbitorului prin intermediul telefonului în mai puțin de 30 de secunde de conversație normală, sistem implementat de Nuance, clienții fiind foarte încântați de funcționalitate, 93% oferind nota 9 din 10 pentru viteză, ușurință de folosire și securitate. De asemenea în cazul modelelor mai noi de telefoane se poate face deblocarea telefonului prin recunoașterea proprietarului [2].

Un motiv pentru care recunoașterea vorbitorului este foarte utilă în anumite domenii este evitarea fraudei, de exemplu în America sistemul de sănătate angajează persoane calificate pentru a se îngriji de oameni ce nu se pot deplasa. Însă nu se puteau face verificări dacă persoanele respective chiar aveau grijă de clienți și astfel s-a ivit un potențial imens de fraudă, foarte mulți îngrijitori aveau "grijă" de foarte multe persoane cu "dizabilități". Soluția care s-a găsit a fost antrenarea îngrijitorilor într-un sistem de recunoaștere de vorbitor pentru ca în momentul în care îngrijitorul ajunge la client și momentul în care pleacă acesta să se autentifice într-un sistem folosind vocea sa. Sistemul de sănătate din America are alocat aproximativ 400 miliarde de dolari anual iar frauda este estimată la aproximativ 30% din angajați deci o sumă foarte mare de bani [3].

Recunoașterea vorbitorului poate fi utilizată în orice domeniu unde este necesară limitarea accesului, de exemplu în instituții guvernamentale, militare sau chiar și în companii.

O altă aplicabilitate a recunoașterii de vorbitor este diarizarea unui flux audio, adică împărțirea și anotarea automată a unei convorbiri pe segmente corespunzătoare fiecărui vorbitor, acest proces fiind de mare utilitate în aplicații de indexare audio / video, putând foarte ușor fi combinată și cu un sistem de recunoaștere de vorbire pentru a creea transcrieri a înregistrărilor [4].

1.3. Starea artei în recunoașterea vorbitorului

Primele încercări de recunoaștere de vorbitor au fost în anii 1960-1970 în cadrul laboratoarelor Bell, unde Pruzansky[13] a fost primul care a inițiat studiul bancurilor de filter și corelarea a două spectrograme digitale pentru a măsura similaritatea. Ulterior bancurile de filtre au fost înlocuite cu analizatoare de formanți de către Doddington în cadrul Texas Instruments. Variabilitatea intravorbitor era cea mai mare problemă în acest moment.[14]

S-a încercat extragerea trăsăturilor de vorbire independent de text prin medierea pe acestora pe perioade suficient de lungi sau prin extragerea statistică sau predictivă a acestora, parametrii extrași fiind autocorelația medie, matricea de covarianță spectrală, histograma frecvenței fundamentale, parametrii LPC și spectrul mediat pe termen lung.

Deoarece performanța sistemelor independente de text era încă limitată au fost investigate și metode în domeniul timp și dependente de text. În cazul metodelor în domeniul timp prin aliniere adecvată se pot face comparații mai precise între două rostiri ale aceluiași text în condiții similare.

Texas Instruments a fost prima companie care a implementat un sistem de verificare vorbitor experimental pe scală largă ce oferă un grad de securitate ridicat. Verificarea se facea pe baza unei înșiruiri de 4 cuvinte monosilabice aleatoare alese dintr-un grup de 16, se foloseau filtre digitale pentru analiză spectrală iar decizia se lua secvențial prin rostirea de 4 ori a acelui enunț.

De asemenea laboratoarele Bell au construit un sistem experimental menit să funcționeze pentru liniile telefonice. Furui a propus utilizarea de coeficienți spectrali împreună cu prima și a doua derivată a acestora pentru a crește robustețea la distorsiuni ale sistemului telefonic, sistemul bazat pe coeficienți

cepstrali devenind standard atât pentru recunoașterea vorbitorului cât și pentru recunoașterea vorbirii[15].

Începând cu anii 1980 s-au dezvoltat sisteme de recunoaștere dependente de text bazate pe modele Markov ascunse (HMM) ca o alternativă la abordarea folosind potrivirea de modele (*template-matching*). Modelele Markov au aceleași avantaje în cazul recunoașterii de vorbitor ca și în cazul recunoașterii de vorbire putându-se obține modele remarcabil de robuste a unor evenimente de vorbire având la dispoziție cantități mici de informații folosite în antrenare.

Ulterior s-au dezvoltat sisteme independente de text bazate pe cuantizare vectorială unde o serie de vectori de caracteristici în timp scurt ale unui vorbitori sunt comprimați într-un set reprezentativ de puncte și stocate într-o așa numită tabelă de cuantizare. De asemenea s-a testat și modelul Markov ascuns ergotic(toate tranzițiile între stări sunt posibile) fiecare rostire fiind caracterizată de un model cu 5 stări în spațiul acustic al caracteristicilor iar Rose et al. au propus utilizarea modelului cu o singură stare cunoscut astăzi sub numele de model cu mixturi Gaussiene (GMM) .

Începând cu anii 1990 cercetările aveau ca scop creșterea robusteții, Matsui et al. [16] au comparat cele două metode din punctul de vedere al robusteții la variația rostirilor și a concluzionat faptul că metoda HMM-urilor ergodice continue este mult superioară metodei discrete și echivalentă metodei bazate pe cuantizare vectorială atunci când este prezentă suficientă informație la antrenare; de asemenea au investigat precizia identificării vorbitorului raportată la numărul de stări și mixturi ale modelului Markov și au descoperit că informațiile legate de tranzițiile între stări sunt irelevante în cazul recunoașterii vorbitorului independente de text GMM-urile obținând o performanță similară cu HMM-urile cu stări multiple.

Metoda textului sugerat propus de Matsui et al. [17]este o metodă de recunoaștere de vorbitor combinată cu recunoașterea de vorbire în care utilizatorul rostește un text generat aleator de fiecare dată când se autentifică iar autentificarea are loc doar când utilizatorul a rostit corect acel text generat aleator. Acest lucru scade precizia datorită sistemului de recunoaștere de vorbire dar crește gradul de securitate deoarece nu se poate autentifica o persoană folosind vocea preînregistrată a utilizatorului.

Cea mai grea problemă în verificarea vorbitorului este normalizarea variației similarității intra-vorbitor, de asemenea apar și diferențe între condițiile de înregistrare, transmisie și zgomot, utilizatorii nu pot rosti în același fel de la o autentificare la alta astfel s-au investigat metode bazate pe rata de probabilitate(*likelihood ratio*) și metode bazate pe *adaptarea a posteriori*. Pentru a reduce costul computațional al termenului de normalizare s-a propus "*metoda cohortelor*" sau "*modelul universal*".

Începând cu anii 2000 s-au propus noi metode de normalizare în care scorurile obținute sunt normalizate prin scăderea mediei și raportarea la deviația standard obținute din distrubuția scorurilor de impostori. Există diferite metode de a calcula distribuția scorurilor de impostori, acestea sunt: Znorm, Hnorm, Tnorm, Htnorm, Cnorm și Dnorm.

Starea artei în verificarea independentă de text este folosirea de una sau mai multe nivele de normalizare (*CMS* și *normalizarea mediei parametrilor cepstrali*) împreună cu o normalizare a modelului universal și unul sau mai multe normalizări are scorurilor.

În prezent s-a observat că supervectorii generați în acest fel nu sunt ideali, *adapatarea a posteriori* nu adaptează doar caracteristicile specifice fiecărui vorbitor, pornind de la modelul universl, el adaptează caracteristicile canalului de comunicații și alți factori perturbatori. Un supervector ar trebui să poată fi descompus în caracteristici specifice vorbitorului, caracteristici dependente de vorbitor, caracteristici dependente de canal și componente reziduale. Aceste componente vor fi discutate în capitolul 2 [5].

1.4. Prezentarea pe scurt a lucrării

Fiind prezentat obiectivul lucrării de a construi un sistem de recunoaștere de vorbitor independent de text prin intermediul comunicației telefonice tindându-se cont de timpul de procesare, variabilitatea inter-vorbitor și robustețea la zgomot lucrarea se structurează astfel:

Capitolul 1 introduce domeniul recunoașterii vorbitorului, ce probleme se ridică și se prezintă pe scurt evoluția sistemelor de recunoaștere de vorbitor de-a lungul timpului.

Capitolul 2 prezintă baza teoretică a sistemului de recunoaștere ce urmează a fi implementat și testat.

Capitolul 3 prezintă pe scurt sistemele de telecomunicații actuale și protocoalele folosite.

Capitolul 4 prezintă contribuția practică, rezultatele obținute , modul de implementare precum și dificultățile întâlnite în implementare.

Capitolul 5 sumarizează principalele concluzii și sublinează contribuția autorului, de asemenea sunt prezentatea direcțiile de dezvoltare ulterioare.

Capitolul 2

Recunoașterea vorbitorului

2.1. Extragerea caracteristicilor vorbitorului

Procesul comun a tuturor formelor de sisteme de recunoaștere de vorbitor și vorbire este extragerea vectorilor de caracteristici din segmente uniform distribuite în timp ale semnalului vocal eșantionat.

2.1.1. Preprocesarea semnalului vocal

Înainte de a extrage caracteristicile semnalul vocal trebuie să sufere următoarele procesări:

a) **Accentuare**: un filtru trece sus este folosit pentru a accentua frecvențele înalte și a compensa faptul că sistemul fonator uman tinde să atenueze aceste frecvențe.
b) **Segmentare:** semnalul vocal este nestaționar în timp lung însă cvasi-staționar în timp scurt, de ordinul 10-30 ms de aceea semnalul vocal este împărțit în segmente de durată fixă numite *cadre.* Dimensiunea tipică a unui cadru este 20 ms ele generându-se din 10 în 10 ms astfel încât să aibă loc o suprapunere de 15 ms de la o fereastră la alta.
c) **Atenuare:** fiecare fereastră este mutiplicată cu o funcție fereastră, de obicei fereastra Hamming, pentru a atenua efectul cauzat de segmentarea cu ferestre finite [6] .

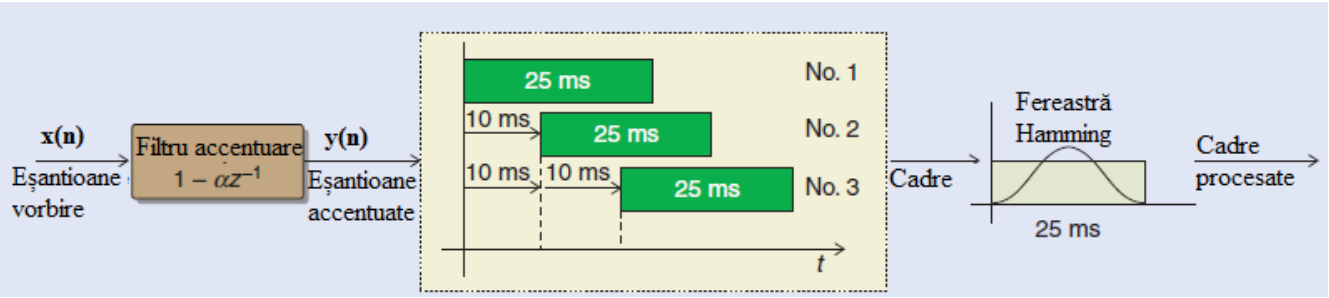

Figura 2.1. Preprocesarea semnalului vocal [6]

2.1.2. Parametrii MFCC

Pentru recunoașterea vorbitorului este important ca din fiecare cadru să se extragă acele caracteristici specifice vorbitorului. Multe astfel de caracteristici au fost investigate, precum coeficienții predicției liniare (LPC), ei fiind derivați direct din modul de producere al vorbirii cât și coeficienții predicției liniare perceptuali (PLP), ei fiind bazați pe sistemul uman de percepție și auditiv.

Cu toate acestea însă în ultimele două decenii caracteristici bazate pe spectru au devenit populare în special datorită faptului că provin direct din transformarata Fourier; Caracteristici bazate pe spectru sunt coeficienții cepstrali în scară Mel (MFCC) și succesul lor se datorează faptului că utilizează un banc de filtre, cu o scară perceptuală similară cu sistemul auditiv uman, pentru a procesa transformata Fourier. De asemenea acești coeficienți prezintă o robustețe la zgomot și flexibilitate datorită procesării cepstrale.

Obținerea coeficienților presupune următoarele procesări :

a) **Transformata Fourier:** se aplică transformata Fourier rapidă (FFT) pentru a obține $N/2$ valori spectrale complexe egal distanțate de la 0 la $Fs/2$. Se ignoră informația despre fază deoarece aceasta diferă foarte mult de la cadru la cadru și se păstrează doar amplitudinea spectrului.

b) **Filtrarea prin bancul de filtre:** cele $N/2$ amplitudini obținute sunt convertite printr-un banc de filtre de ordin K pentru a reduce dimensiunea și pentru a reprezenta mai eficient coeficienții, de asemenea filtrarea se poate face logaritmic în loc de linear din motive perceptuale. Bancul de filtre este astfel proiectat încât să diferențieze mai bine frecvențele joase și să grupeze cât mai multe frecvențe înalte, centrele filtrelor triunghiulare sunt distanțate conform formulei:

$$f_{MEL} = 2595 \log_{10}\left(1 + \frac{f_{LIN}}{700}\right) \tag{2.1}$$

ieșirea fiecărui filtru este notată cu $S_k, k = 1, 2, \dots, K$, iar datorită faptului că sistemul auditiv are caracter logaritmic se logaritmează această ieșire, în plus logaritmarea are și rolul de a transforma distorsiunile multiplicative, provocate de filtrarea în frecvență, în efecte aditive ce pot fi compensate mai ușor.

c) Analiza cepstrală – pasul final este de a converti cele K valori spectrale de la ieșirea filtrului $\{\log(S_k)\}_{k-1}^{K}$ în L coeficienți cepstrali folosind transformata cosinus discretă (DCT) :

$$c_n = \sum_{k=1}^{K} \log(S_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \quad \text{n} = 1, 2, \dots, \text{L} \tag{2.2}$$

De obicei doar $L = 12$ coeficienți sunt suficienți pentru fiecare fereastră de analiză.

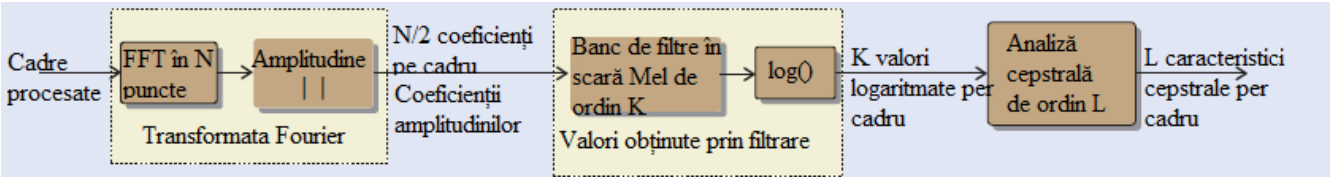

Figura 2.2. Obținerea coeficienților MFCC [6]

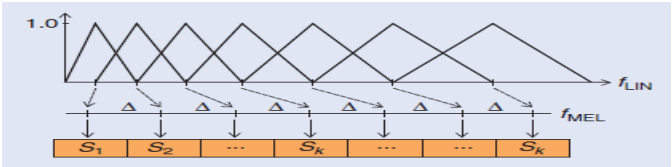

Figura 2.3. Bancul de filtre în scară Mel [6]

2.1.3. Compensarea zgomotului

Principalul avantaj al transformării amplitudinii spectrului în coeficienți cepstrali logaritmați este transformarea efectului multiplicativ al zgomotului de canal și ambiental(mai ales de la utilizarea de microfoane diferite în procesul de antrenare și testare) într-un efect aditiv. Presupunând că aceste efecte sunt invariante pe durata rostirii se poate aplica o procedură simplă de scădere a mediei coeficienților cepstrali totali în fiecare fereastră. Ideea este de a înlătura orice efect invariant în timp împreună cu informația medie de vorbire, lăsând în urmă doar variațiile importante, dinamice, de vorbire care caracterizează cel mai bine vorbitorul.

Acești coeficienți compensați sunt obținuti prin procesul de normalizare a mediei cepstrale (CMN) precum urmează:

1) Calcularea mediei peste toți coeficienții MFCC ale rostirii:

$$\vec{\mu}_T = \frac{1}{T}\sum_{t-1}^{T} \vec{c}_t \tag{2.3}$$

unde $\vec{c}_t = [c_1, c_2, \dots, c_L]_t$ este vectorul de coeficienți MFCC ai ferestrei de index t și sunt T ferestre în total.

2) Compensarea vectorului de coeficienți MFCC ai fiecărei ferestre :

$$\vec{c}_t^c = \vec{c}_t - \vec{\mu}_T \tag{2.4}$$

În verificarea vorbitorului această preprocesare este sporită de strategii de normalizare ale canalului iar în identificarea vorbitorului alte strategii de normalizare a mediului pot fi aplicate în special dacă antrenarea se face în medii diferite, de asemenea normalizarea se poate face și prin raportarea la varianța mediului.

$$\vec{c}_t^c = (\vec{c}_t - \vec{\mu}_T)/\vec{\sigma}_e \tag{2.5}$$

2.1.4. Concatenarea derivatelor temporale

Prin derivarea coeficienților cepstrali se poate observa dinamica vorbirii, acest aspect joacă un rol important în recunoașterea vorbitorului ajutând la identificarea stilului și durata vorbirii astfel :

$$\vec{d}_t = \frac{\sum_{p=1}^{P}\left(\vec{c}_{t+p}^c - \vec{c}_{t-p}^c\right)}{2\sum_{p=1}^{P} p^2} \tag{2.6}$$

unde $\vec{c}_t^c$ reprezintă vectorul compensat al ferestrei t iar P de obicei este 2, derivarea făcându-se între coeficienți succesivi în cadrul ferestrei.

Prin derivarea încă o dată se obțin derivata de ordinul doi $\vec{a}_t$, numită și *accelerația* parametrilor.

Aceste derivate temporale sunt concatenate coeficienților MFCC orginali pentru a obține un vector de caracteristici format din coeficienții MFCC + delta + accelariția de dimensiune 39 precum în figura 2.4 [6].

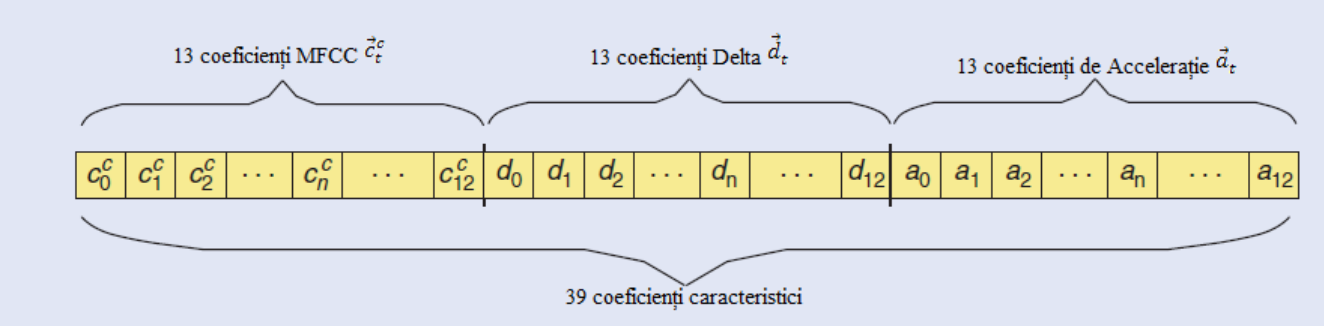

Figura 2.4. Vectorul de caracteristici [6]

2.2 Modelarea vorbitorului folosind sisteme GMM și GMM-UBM

2.2.1. Modele bazate pe mixturi Gaussiene (GMM)

Presupunând o rostire formată din *T* cadre pentru un vorbitor *j* și un vector de caracteristici de lungime *D* pentru fiecare cadru adică pentru fiecare rostire avem :

$$\vec{x}_t \epsilon R^D : 1 \leq t \leq T \tag{2.7}$$

dacă se poate contrui un model al vorbitorului *j* astfel încât pentru orice rostire a vorbitorului din acele *T* cadre să se poată reprezenta vorbitorul *j* și nu alt vorbitor atunci putem recunoaște corect vorbitorul.

Luând în considerare faptul că rostirile unui vorbitor sunt formate din secvențe aleatoare în timp cuprinzând toate posibilitățile de cuvinte rostite cel mai bun candidat pentru a face recunoașterea este un model statistic, astfel cea mai generică metodă de modelare este folosirea de modele cu mixturi Gaussiene.

Un model Gaussian presupune faptul că vectorul de caracteristici urmează o lege Gaussiană caracterizată de medie și deviație standard, permițând utilizarea unui număr mai mare de Gaussiene se poate caracteriza distribuția caracteristicilor unui vorbitor particular mai bine.

Modelul de mixturi Gaussiene a unui vorbitor *j*, λ_j , este o sumă ponderată a *M* componente de densități generate cu formula:

$$p(\vec{x}_t|\lambda_j) = \sum_{i=1}^{M} g_i \mathcal{N}(\vec{x}_t; \vec{\mu}_i; \Sigma_i). \tag{2.8}$$

unde g_i reprezintă ponderile mixturilor ce satisfac condiția $\sum_{i=1}^{M} g_i = 1$ iar $\mathcal{N}(\vec{x}_t; \vec{\mu}_i; \Sigma_i)$ reprezintă componentele de densități individuale ce pentru un vector de caracteristici *D* dimensional au forma din ecuația 2.9

$$\mathcal{N}(\vec{x}_t; \vec{\mu}_i; \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\vec{x}_t - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x}_t - \vec{\mu}_i)} \quad (2.9)$$

cu media $\vec{\mu}_i \epsilon R^D$ și matricea de covarianță $\Sigma_i \epsilon R^{D \times D}$. Modelul cu mixturi Gaussiene pentru un vorbitor *j*, λ_j, este parametrizat prin vectorii de medie, matricea de covarianță și ponderile mixturilor Gaussiene ale tutoror celor *M* componente de densități precum în figura 2.5.

$$\lambda_j = \{\vec{\mu}_i, \Sigma_i, g_i\}_j \qquad i = 1, 2, \ldots, M. \quad (2.10)$$

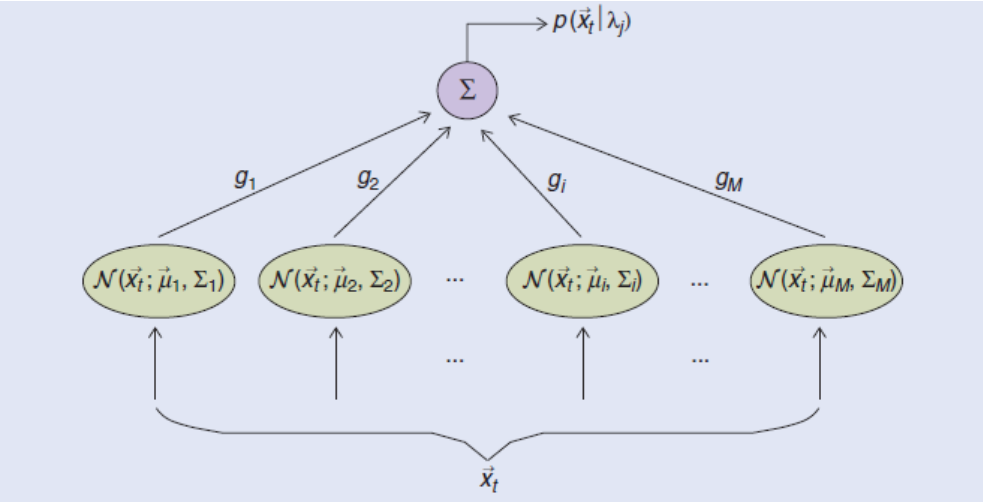

Figura 2.5. Reprezentarea mixturilor Gaussiene [6]

Acest model trebuie să colecteze în mod imparțial toate informațiile de vorbire ale unui vorbitor și să modeleze toate posibilitățile de variabilități acustice ale vorbitorului, având la dispoziție suficiente mixturi (în jur de 64 sau mai mult) se poate face o reprezentare corectă a vastei distribuții de foneme pronunțate. Un instrument matematic este algoritmul de maximizare a probabilităților (*Expectation-Maximization*) ce garantează convergența spre un set optim de parametri în doar câteva iterații.

Dezavantajul acestui algoritm este faptul că necesită un număr suficient de informații pentru antrenare și în cazul în care la testare apare informație neexistentă în antrenare va genera un scor mic degradând performanța sistemului, soluția fiind existența unui set de antrenare cât mai diversificat [6].

2.2.2. Modelul universal de mixturi

În verificarea vorbitorului identitatea se verifică prin compararea unei rostiri cu modelul stocat dar și cu modelul impostorului. Modelul impostorului este un model cu mixturi Gaussiene care modelează toți vorbitorii în afară de vorbitorul dorit și poartă denumirea de model universal (UBM); deși acest model ar trebui să conțină mixturile tuturor vorbitorilor mai puțin ale celui dorit în practică se stochează toate mixturile chiar și ale vorbitorului dorit având astfel avantajul de a putea fi folosit pentru orice verificare. Un astfel de model, datorită numărului foarte mare de vorbitori, în mod uzual este format dintr-un număr mai mare de distribuții Gaussiene, de ordin 256 sau mai mult, el reprezentând distribuția independentă de vorbitor a întregii populații.

Deși un model universal poate fi folosit pe un set deschis de vorbitori pentru a detecta vorbitorii necunoscuți într-un set închis de vorbitori nu este neapărat nevoie utilizarea unui model universal, modelele fiecărui vorbitor fiind suficiente pentru identificare, cu toate acestea un model universal este mai bine antrenat decât orice model cu mixturi Gaussiene deoarece modelează toate caracteristicile ale tuturor vorbitorilor și nu suferă de probleme precum insuficiente date de antrenare. În plus datorită procesului de maximizare a probabilităților cu puține informații se poate adapta un model al unui vorbitor pornind de la modelul universal folosind adaptarea *Maximum A-Posteriori*; astfel este mai ușor să contruiești un model general și apoi prin MAP să extragi modelul unui vorbitor specific precum în figura 2.6 [6].

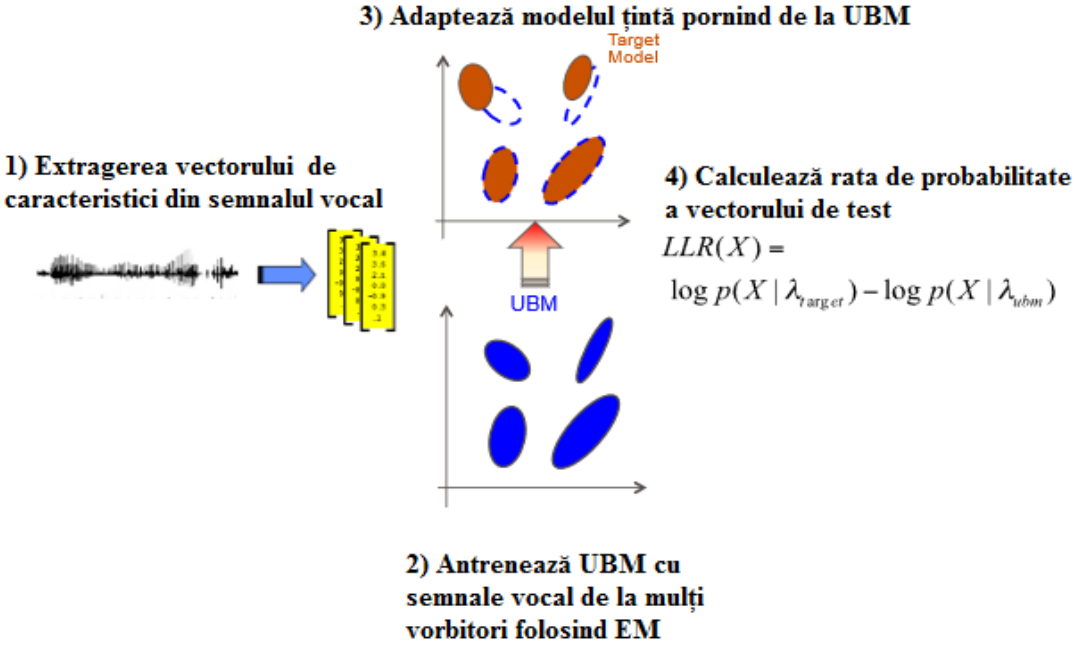

Figura 2.6. Antrenarea și testarea folosind modelul universal [8]

2.3. Modelare Joint Factor Analysis

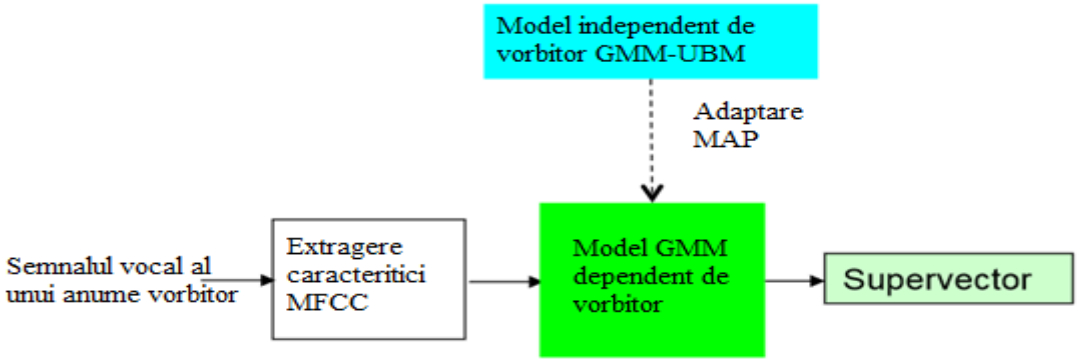

Figura 2.7. Schema bloc de obținere a unui supervector [7]

În recunoașterea vorbitorului cu ajutorului modelului universal se poate adapta un model specific unui vorbitor numit supervector, acest lucru se face prin adaptare MAP și este o interpolare liniară a tuturor componentelor de mixturi Gaussiene ale modelului universal pentru a crește probabilitatea logaritmică a vorbirii unui anumit vorbitor. Supervectorii conțin toate componentele mediate ale mixturilor Gaussiene dependente de vorbitor [7].

Problema supervectorilor este că sunt adaptați atât la caracteristicile specifice ale unui vorbitor dar și la zgomotul de canal sau alți factori perturbatori, în acest sens supervectorii nu sunt ideali, un supervector ar trebui să poată fi descompus în informații dependente de vorbitor, informații independente de vorbitor, informații dependente de canal și elemente reziduale astfel:

$$s = m + Vy + Ux + Dz \tag{2.11}$$

unde :

- *m* este vectorul independent de vorbitor
-*V* este matricea proprie de voce
- *y* este vectorul de factorizare a vorbitorului
- *U* matricea proprie a canalului
- *x* este vectorul de factorizare a canalului
- *D* este matricea reziduală și este diagonală
- *z* este vectorul de factorizare a componentelor reziduale dependente de vorbitor

Fiecare componentă poate fi reprezentată printr-un set redus dimensional de factori care operează conform cu dimensiunile principale ale componentei corespunzătoare (dimensiuni proprii).

Un vector propriu este un vector nenul ce multiplicat cu o matrice de ordin $N \times N$ va rezulta într-o constantă, numită valoare proprie, multiplicată cu vectorul respectiv:

$$Av = \lambda v \tag{2.12}$$

unde A este matricea, v este vectorul iar λ este valoarea propie.

De exemplu:

$$V * y = \begin{bmatrix} | & | & | & | \\ v_1 & v_2 & \cdots & v_N \\ | & | & | & | \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{2.13}$$

unde *V* este matricea de voce propie, *y* reprezintă vectorul de dimensiune redusă ai vorbitorului, fiecare factor controlând o coloană din matricea de voce *V* [7].

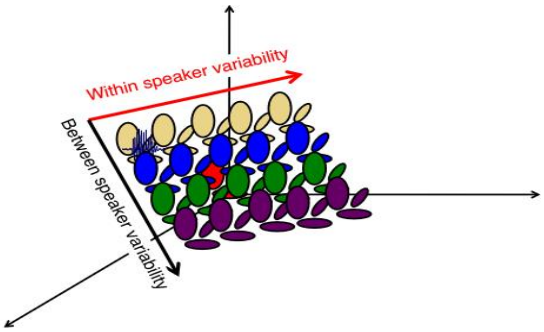

Figura 2.8. Spațiul variabilității

În figura 2.8 este reprezentat modul în care sunt antrenați vorbitorii, analiza conține atât informații despre canal cât și componente reziduale.

2.4. Vectorii intermediari. *I-vectors*

S-a observat că în cazul modelării JFA variabilitățile matricii V și matricii U sunt similare, canalul de vorbire conținând informații despre vorbitor, astfel s-a propus crearea unui sistem intermediar care să conțină ambele variabilități iar compensarea canalului să se facă ulterior în spațiul vectorilor intermediari:

$$s = m + Tw \tag{2.14}$$

unde *T* reprezintă matricea de variabilitate totală iar *w* vectorul standard intermediar, *w* se presupune a avea o distribuție de medie 0 și varianță 1;

2.4.1. Extragerea vectorilor intermediari

Sistemul de extragere a i-vectorilor este caracterizat de :

-*m*: mediile modelului universal de mixturi;

-*T*: matricea de variabilitate de rang mic;

-∑: matricea diagonală de covarianță;

De asemenea se dau numărul de cadre de vorbire $u = \{\vec{x}_1, \dots, \vec{x}_L\}$ unde fiecare $\vec{x}_t$ este de dimensiune *F*; numărul de componente Gaussiene *C* indexate cu ajutorului *c*, dimensiunea supervectorului fiind $C * F$;

Primul pas în extragerea i-vectorilor este procesearea Baum-Welch statistică pentru a extrage dependența dintre vorbitori.

Reamintim:

$$\gamma_t(c) = P(c|\vec{x}_t, \theta_{UBM}) = \frac{g_c P_c(\vec{x}_t|\mu_c, \Sigma_c)}{\sum_{i=1}^{C} g_i P_i(\vec{x}_t|\mu_i, \Sigma_i)} \tag{2.15}$$

Procesarea de ordinul zero:

$$N_c(u) = \sum_{t=1}^{L} P(c|\vec{x}_t, \theta_{UBM}) = \sum_{t} \gamma_t(c) \tag{2.16}$$

pentru vorbitorul *u* se însumează toate componentele Gaussiene la momentul de observație *t.*

Procesarea de ordinul întâi:

$$F_c(u) = \sum_{t=1}^{L} P(c|\vec{x}_t, \theta_{UBM}) \cdot \vec{x}_t \ = \sum_{t} \gamma_t(c) \cdot \vec{x}_t \tag{2.17}$$

Procesarea de ordinul doi:

$$S_c(u) = diag\left(\sum_{t} \gamma_t(c) \cdot \vec{x}_t \vec{x}_t^{\ t}\right) \tag{2.18}$$

Centrarea statisticilor de ordinul 1 și 2

$$\tilde{F}_c(u) = \sum_{t} \gamma_t(c) \cdot (\vec{x}_t - m_c) \tag{2.19}$$

$$\tilde{S}_c(u) = diag(\sum_{t} \gamma_t(c) \cdot (\vec{x}_t - m_c)(\vec{x}_t - m_c)^t) \tag{2.20}$$

unde m= $[m_1\ ,\ m_2, \dots, m_c]^t$ reprezintă mediile mixturii universale pentru fiecare componentă *c.*

Prin expansiune matricile arată astfel:

$$\mathrm{N(u)} = \begin{bmatrix} N_1(u) \cdot I_{F\times F} & 0 & \cdots & 0 \\ 0 & N_2(u) \cdot I_{F\times F} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \vdots & 0 & N_c(u) \cdot I_{F\times F} \end{bmatrix} \tag{2.21}$$

$$\tilde{F}(u) = \begin{bmatrix} \tilde{F}_1(u) \\ \tilde{F}_2(u) \\ \vdots \\ \tilde{F}_C(u) \end{bmatrix} \tag{2.22}$$

$$\tilde{S}(u) = \begin{bmatrix} \tilde{S}_1(u) & 0 & \cdots & 0 \\ 0 & \tilde{S}_2(u) & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \vdots & 0 & \tilde{S}_c(u) \end{bmatrix} \quad (2.23)$$

Aceste matrici sunt salvate spre a fi utilizate ulterior, urmează un proces de maximizare a probabilităților în care se inițializează *m* și $\sum$ cu valorile definite în modelul universal , de asemenea se inițializează o matrice de dimensiune $CF \times R$ aleatoare unde *R* este rancul dorit din matricea *T*.

În pasul *E* pentru orice rostire *u* se calculează parametrii ale distribuției viitoare de vectori $w(u)$ folosind valorile curente ale lui *m*, *T* și $\sum$ astfel:

fiind definită matricea $l(u) = I + T^t\Sigma^{-1}\mathrm{N}(u)T$ (2.24)

se poate calcula distribuția următoare a lui $w(u)$:

$$E[w(u)] = l^{-1}(u)T^t\Sigma^{-1}\tilde{F}(u) \quad (2.25)$$

și matricea de covarianță:

$$cov\big(w(u), w(u)\big) = l^{-1}(u) \quad (2.26)$$

În pasul *M* se actualizează matricea *T* și $\sum$ prin rezolvarea de ecuații liniare în care $w(u)$ joacă rolul de soluții.

$$N_c = \sum_u N_c\,(u) \quad (2.27)$$

$$A_C = \sum_u N_c(u)E[w(u)w^t(u)] \quad (2.28)$$

$$C = \sum_u \tilde{F}(u)E[\,w^t(u)] \quad (2.29)$$

$$N = \sum_u N(u) \quad (2.30)$$

$$E[w(u)w^t(u)] = Cov\big(w(u), w(u)\big) + E[w(u)] \cdot E[w^t(u)] \quad (2.31)$$

$$T(i,:)A_c = C_i \quad (2.32)$$

$$\Sigma = \mathrm{N}^{-1}(\sum_u \tilde{S}(u) - diag(CT^t) \quad (2.33)$$

unde $i = (c-1) * DF$ iar *DF* reprezintă dimensiunea vectorului de caracteristici.

Se repetă pașii *E* și *M* de aproximativ 20 de ori până când probabilitatea converge , valoarea finală a lui *w(u)* fiind i-vectorul.

Compensarea canalului în cazul i-vectorilor se face prin tehnici precum analiza lineară (LDA) urmate de WCCN , i-vectorii compensați fiind notați ω.

Pentru a obține scorul se aplică măsurarea distanței cosinului asupra i-vectorilor antrenați și cei testați:

$$scor(\omega_1, \omega_2) = \frac{{\omega_1}^* * \omega_2}{\|\omega_1\| * \|\omega_2\|} = \cos\left(\theta_{\omega_1,\omega_2}\right) \tag{2.34}$$

Dacă doi i-vectori sunt similari tind spre valoarea 1 în sens contrar tind spre valoare -17 [8].

2.4.2. Compensarea canalului

Vectorii intermediari sunt extrași astfel încât nu se poate face distincție între variabilitatea canalului de comunicații și variabilitatea vorbitorilor. Astfel, această problemă trebuie rezolvată prin construirea de clasificatori folosind i-vectorii drept caracteristici. S-au propus două metode LDA (*Linear Discriminant Analysis*) și WCCN(*Within Class Channel Normalization).*

a) Analiza liniară

Analiza LDA este o metodă supervizată cu scopul de a reduce dimensionalitatea, puterea acestei metode șade în faptul că folosește etichetele claselor pentru a găsi spațiul optim minim.

Această optimizare pe lângă faptul că reduce spațiul din interiorul unei clase ajută și la maximizarea variabilității intre clase prin optimizarea funcției Fisher:

$$J(u) = \frac{u^t S_b u}{u^t S_w u} \tag{2.35}$$

unde S_b și S_w reprezintă matricea de covarianță dintre clase și matricea de covarianță din interiorul clasei, *u* fiind direcția de orientare din spațiu.

Fiind date un set de antrenare de *S* vorbitori cu n_s rostiri ale fiecărui vorbitor, funcția de optimizare Fisher va încerca să rezolve sistemul propiu generalizat:

$$S_b u = \lambda S_w u \tag{2.36}$$

unde
$$S_b = \sum_{j=1}^{S}\left(w_j - \overline{w}\right)\left(w_j - \overline{w}\right)^t \tag{2.37}$$

$$S_w = \sum_{j=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}({w_i}^s - \overline{w}_s)({w_i}^s - \overline{w}_s)^t \tag{2.38}$$

${w_i}^s$ reprezintă observația *i* a vorbitorului *s*, $\overline{w}_s = \frac{1}{n_s}\sum_{i=1}^{n_s} {w_i}^s$ este media observațiilor vorbitorului *s* iar $\overline{w}$ reprezintă media tuturor instanțelor din setul de antrenare.

În cazul i-vectorilor media $\overline{w}$ este nulă deoarece s-a făcut prezumția că i-vectorii sunt distribuiți normalizat de medie 0 și matricea de covarianță identitate.

b) Normalizarea covarianței din interiorul clasei

Această metodă a fost introdusă în contextul clasificării cu SVM și propune utilizarea inversei matricii de covarianță din interiorul clasei pentru a normaliza nucleul liniar.

$$W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(A^t w_i^s - w_s)\,(A^t w_i^s - w_s)^t \tag{2.39}$$

Matricea *A* este dată de vectorii proprii asociați valorilor proprii cele mai mari din ecuația 2.39.

2.4.3. Normalizarea vectorilor intermediari

Cele mai bune performanțe s-au obținut combinând metodele LDA și WCCN, deoarece analiza liniară elimină efectul liniar al canalului împreună cu reducerea dimensionalității iar WCCN elimină efectul radial al canalului. Se dorește să se elimine efectul radial al canalului înainte de reducerea dimensionalității astfel vectorii trebuie să fie standardizați și normalizați. Deoarece i-vectorii ar trebui să aibă media 0 și varianța 1 diferența dintre valorile reale și valorile dorite poate fi ignorată astfel i-vectorii pot fi standardizați. În plus efectul radial poate fi eliminat prin normalizarea lungimilor.

Fie $\overline{w}$ și *V* matricile de medii și covarianță ale unui set de vectori intermediari folosiți în antrenare. Pentru a aplica standardizarea și normalizarea lungimii se descompune matricea *V* în PDP^t unde *P* reprezintă vectorul propiu al matricii *V* iar *D* este matricea *V* diagonalizată.

Un vector w de antrenare este transformat în w' astfel:

$$w' = \frac{D^{-\frac{1}{2}}P^t(w - \overline{w})}{\sqrt{(w - \overline{w})^t V^{-1}(w - \overline{w})}} \tag{2.40}$$

Același proces se aplică și vectorilor de testare folosind setul de antrenare $\overline{w}$ și *V*.

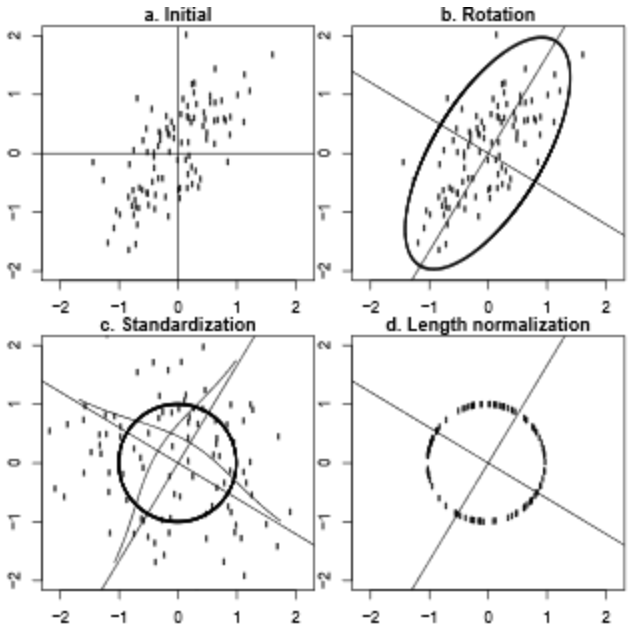

Figura 2.9. Exemplu normalizare și standardizare

După cum se vede în figura 2.9 în a) este reprezentată forma inițială a vectorilor, aplicând P^t se aplică o rotație în jurul axelor principale ale spațiului de variabilitate totală b) apoi prin aplicarea lui $D^{-\frac{1}{2}}$ aceștia se distribuie unirform c) iar prin normarea lungimii aceștia ajung în același plan al sferei d)[8] g.

2.5. Sisteme biometrice

2.5.1. Arhitectura unui sistem biometric

Un sistem biometric este un sistem de recunoaștere unică a persoanelor pe baza uneia sau mai multor trăsături intrinseci fizice sau comportamentale, numite date biometrice. Biometria are o aplicabilitate foarte mare în securitate, ea putând fi folosită pentru a autentifica identitatea unei persoane în scopul de a permite accesul la zone restricționate sau la sisteme electronice bazându-se pe premiza că trăsăturile fizice examinate sunt capabile să identifice în mod unic persoana respectivă [1].

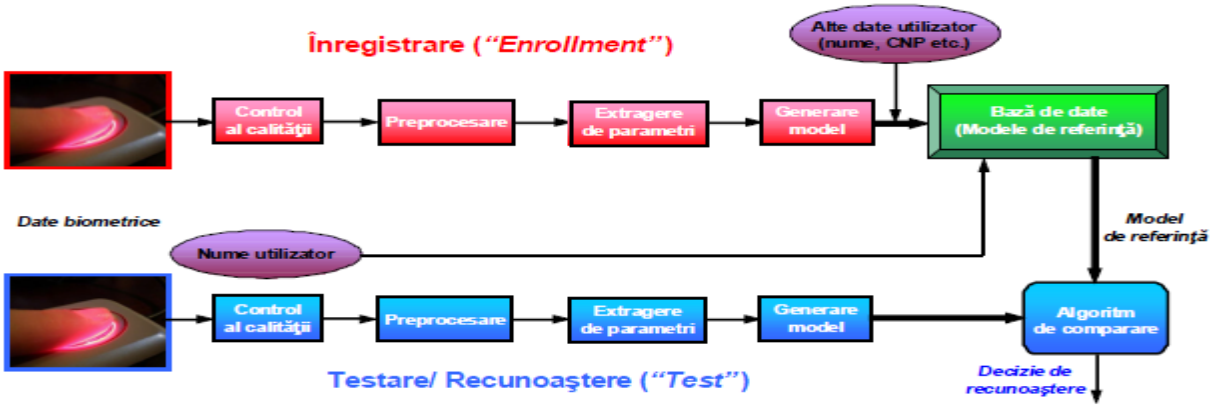

Figura 2.10. Structura unui sistem biometric [19]

Un sistem biometric trebuie să parcurgă două etape. O etapă inițială numită antrenare sau înrolare în care trăsătura biometrică este extrasă cu ajutorul unui senzor, se generează un model ce este stocat în baza de date împreună cu un identificator al utilizatorului. În cea de a doua etapă numită testare sau autentificare, se repetă pașii de la antrenare cu excepția faptului că modelul extras este comparat cu modelele stocate generând un scor care dacă trece de un prag stabilit empiric poate autentifica sau respinge utilizatorul.

Procesul de autentificare presupune recunoașterea identității unui utilizator însă termenul de recunoaștere se poate clasifica în două mari categorii:

a) Verificare – presupune confirmarea identității susținute a unei persoane ce dorește accesul în sistem prin compararea probei biometrice prezentate cu cea stocată în baza de date. Verificarea presupune o comparare unul la unul.

b) Identificare – presupune descoperirea identității persoanei ce dorește accesul în sistem prin compararea probei biometrice prezentate cu toate modelele stocate în baza de date, procesul de identificare fiind astfel mult mai lent. Identicarea presupune o comparare unul la mai mulți.

Sistemele biometrice pot fi analizate din punct de vedere al **universalității** trăsăturilor.Trăsăturile trebuie să fie comune tuturor oamenilor, de asemenea ele trebuie să fie **unice**(să nu existe două persoane cu aceeași trăsătură). Un alt factor este **permanența** trăsăturii(nu trebuie ca aceasta să se modifice în timp sau să fie alterată) și nu în ultimul rând trăsătura trebuie să fie ușor **măsurabilă** și **cuantificabilă** cu ajutorul unui senzor, de asemenea este de preferat ca metoda de măsurare a trăsăturii să fie **neintruzivă**.

Există o mulțime de categorii de trăsături fizice ce se pot măsura precum amprenta digitală, geometria mâinii, retina, irisul, fața, scrisul și nu în ultimul rând vocea. Fiecare metodă are avantajele și dezavantajele ei, nici un sistem biometric nu este capabil să ofere acuratețe maximă. De exemplu în cazul scanării degetelor se extrag discontinuitățile amprentei și se stochează un model spre a fi folosit pentru comparație la autentificare. Pentru a crește precizia se pot măsura amprentele de la mai multe degete. Cu toate acestea sunt o serie de factori care pot afecta autentificarea, prezența mizeriei atât pe deget cât și pe senzor, îmbătrânirea, uzarea pielii degetului.

În cazul autentificării prin voce există o serie de avantaje precum costul redus al implementării, senzorul de recepție poate fi un microfon sau chiar un telefon, spre deosebire de alte trăsături ce necesită senzori speciali care pot fi fie scumpi și chiar intruzivi cum ar fi cazul scanării retinei.

Autentificarea prin voce este ușor de folosit și universal acceptată. Vvorbitul este un proces natural spre deosebire de poziționarea ochiului în fața unui senzor. De asemenea recunoașterea de vorbitor are avantajul de a putea fi folosită la distanță, este mult mai ușor să permiți unui utilizator să se autentifice folosind vocea prin telefon în scopul de a face un transfer bancar decât să îl rogi să vină la un sediu al băncii și să se autentifice folosind semnătura sau amprenta.

Alt avantaj ar fi faptul că înrolarea se face rapid, utilizatorul este rugat să vorbească un anumit set de cuvinte sau un anumit timp, 2-8 secunde de vorbire fiind suficiente pentru a reprezenta digital amprenta vocală. De asemenea autentificarea se poate face foarte rapid prin compararea cu amprenta vocală înrolată. În plus spațiul de stocare al amprentei vocale este foarte mic aceasta putând fi stocată chiar și pe telefonul utilizatorului.

Dezavantajul amprentei vocale constă în faptul că nu este cel mai sigură trăsătură biometrică. Comparat cu scanarea irisului sau ale retinei oferă un grad de siguranță mai mic, motiv pentru care ar trebui să fie folosit în concordanță cu alte metode de autentificare precum parolele, în plus vocea oamenilor diferă în timp, vocea se poate schimba și datorită stării de sănătate, stresului sau oboselii[1].

Characteristic	Fingerprints	Hand Geometry	Retina	Iris	Face	Signature	Voice
Ease of Use	High	High	Low	Medium	Medium	High	High
Error incidence	Dryness, dirt, age	Hand injury, age	Glasses	Poor Lighting	Lighting, age, glasses, hair	Changing signatures	Noise, colds, weather
Accuracy	High	High	Very High	Very High	High	High	High
Cost	*	*	*	*	*	*	*
User acceptance	Medium	Medium	Medium	Medium	Medium	Medium	High
Required security level	High	Medium	High	Very High	Medium	Medium	Medium
Long-term stability	High	Medium	High	High	Medium	Medium	Medium

Tabelul 1.1 - Comparație sisteme biometrice [1]

2.5.2. Acuratețea unui sistem biometric

Acuratețea unui sistem biometric este afectată de o mulțime de factori chiar și sistemul telefonic folosit poate fi problematic, de exemplu înrolarea unui utilizator folosind telefonul fix și autentificarea acestuia folosind telefonul mobil poate fi suficient pentru a cauza falsa respingere. De asemenea zgomotul de fundal este un factor important. Putem împărți măsurarea acurateții în trei categorii:

- Failure to enroll – înrolarea utilizatorului nu a fost îndeplinită
- False acceptance – utilizatorul a fost autentificat deși nu trebuia
- False rejection – utilizatorul nu a fost autentificat deși ar fi trebuit

Un sistem biometric bun trebuie să aibă un echilibru între rata de acceptare falsă(FAR) și rata de rejecție falsă(FRR). Alterând procesul de decizie astfel încât să scadă rata de falsă acceptare poate duce la creșterea ratei de falsă rejecție și vice-versa. În figură se observă că locul de intersecție a celor două se numește *Crossover Error Rate*(CER) sau *Equal Error Rate*(EER). Cu cât pragul de intersecție este mai mic cu atât precizia sistemului este mai bună[1].

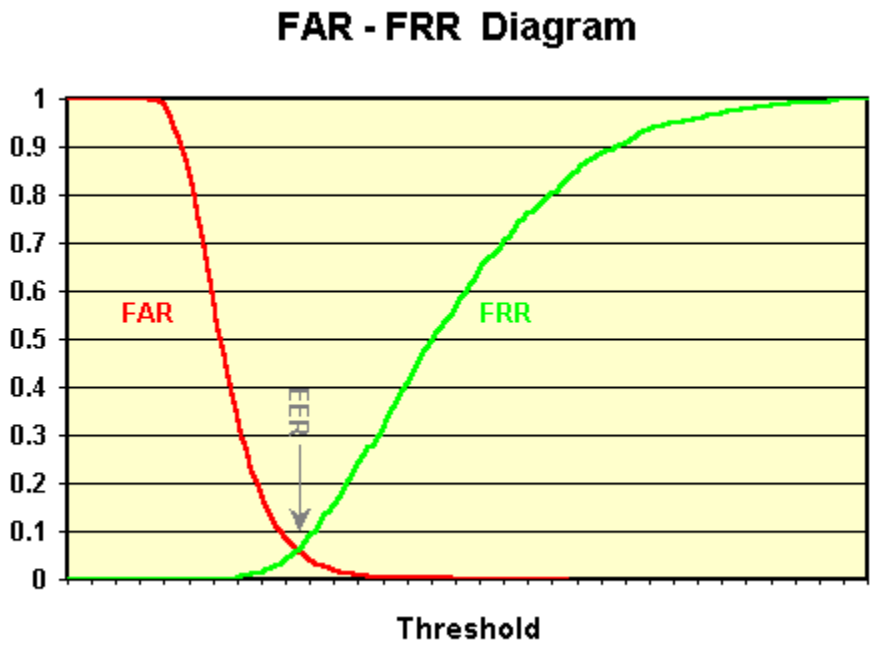

Figura 2.11. Crossover Error Rate [12]

Rata de respingere falsă poate fi definită ca raportul dintre numărul de respingeri false și numărul de total de încercări valide de autentificare ale unui utilizator autorizat.

Rata de acceptare falsă poate fi definită ca raportul dintre numărul de acceptări false și numărul total de încercări de autentificare ale unui impostor.

Acestea fiind spuse putem defini :

fn – numărul de respingeri false – ***f**alse **n**egatives*

fp – numărul de acceptări false – ***f**alse **p**ositives*

tn – numărul de respingeri corecte – ***t**rue **n**egatives*

tp – numărul de acceptări corecte – ***t**rue **p**ositives*

$$FRR = \frac{fn}{tp+fn} \tag{2.41}$$

$$FAR = \frac{fp}{tn+fn} \tag{2.42}$$

Pentru a optimiza pragul de decizie se poate reprezenta grafic raportul între FRR și FAR pentru diferite valori ale pragului. Această reprezentare grafică poarte numele de curbă ROC(*Receiver Operating Characteristic*) și se observă în figura 2.12 că diagonala reprezintă EER iar cazurile optime sunt cele aflate deasupra diagonalei principale. [19]

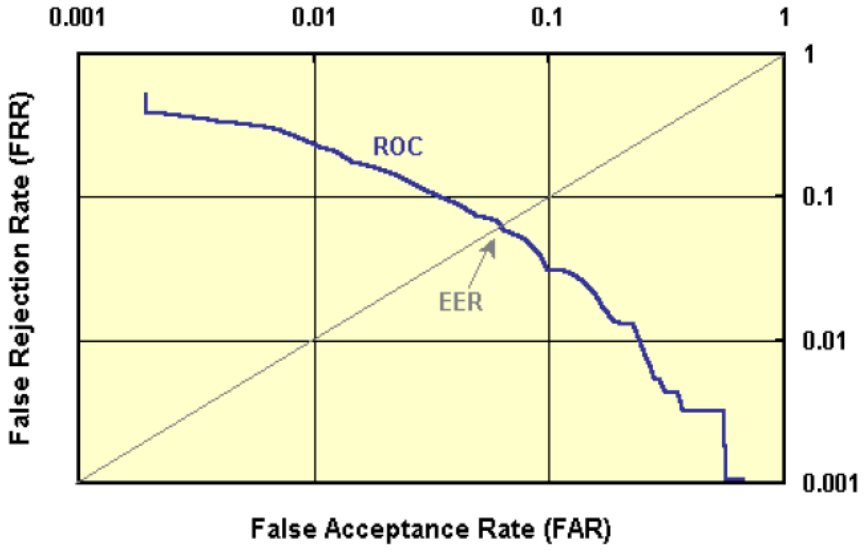

Figura 2.12. Curba ROC[12]

Trebuie remarcat faptul că rata de FAR poate fi ajustată în algoritmul de recunoaștere prin modificarea pragului, cu cât este pragul mai mare cu atât scade rata de acceptare falsă însă acest lucru aduce cu sine și creștere a ratei de falsă respingere. Scopul este e a avea o rata de acceptare falsă cât mai mică oricare ar fi valoarea ratei de falsă respingere și vice versa. De asemenea trebuie luat în considerare și gradul de securitate necesar, în unele cazuri e de preferat ca utilizatorul să repete încercarea de autentificare decât să permită sistemul autentificarea unui impostor.

Rata de falsă respingere poate fi îmbunătățită fie prin utilizarea unui senzor mai bun (în cazul nostru un microfon sau autentificarea într-o zonă cu mai puțin zgomot), fie prin creșterea numărului de înrolări în sistem sau prin îmbunătățirea algoritmului de recunoaștere.

,

Capitolul 3

Sisteme de telecomunicații prin voce

Sistemele de telecomunicații prin voce cuprind comunicarea prin rețelele de telefonie fixă, comunicarea prin rețelele mobile și comunicarea prin sistem de tipul Voice Over IP.

În cadrul lucrării de disertație recunoașterea de vorbitor trebuie să fie făcută în timpul sau în urma convorbirii dintre un client și un operator. Comunicarea se face folosind o centrală telefonică Asterisk ce are la bază protocolul SIP(*Session Initiation Protocol*).

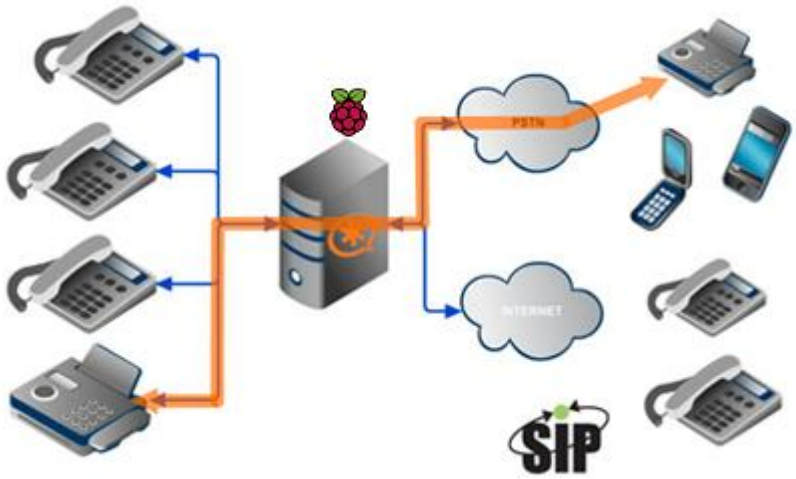

Figura 3.1. Structura unui sistem de telecomunicații

3.1. Session Initiation Protocol

Sistemele de telecomunicații actuale se bazează pe SIP (session initiation protocol) , un protocol de control al nivelului aplicație pentru crearea, modificarea și încheierea sesiunilor de telecomunicații pentru unul sau mai mulți participanți în cadrul telefoniei IP, distribuției multimedia și conferințelor multimedia.

Protocolul SIP a fost proiectat să fie independent de nivelul de transport din substrat, el putând rula pe Transmission Control Protocol (TCP), User Datagram Protocol(UDP) sau Stream Control Transmission Protocol (SCTP), de asemenea SIP este un protocol text-based incorporând elemente ale Hypertext Transfer Protocol (HTTP) și Simple Mail Transfer Protocol (SMTP).

Protocolul funcționează în conjuncție cu diferite protocoale ale nivelului aplicație care identifică și transportă streamurile media precum *Session Description Protocol* (SDP) ce are rolul de a identifica și negocia transferul, *Real-time Transport Protocol* (RTP) sau *Secure Real-time Transport Protocol* (SRTP) ce are rolul de a transmite streamurile de voce sau video, criptarea făcându-se prin *Transport Layer Security* (TLS).

Proiectarea SIP conține elemente de cerere/răspuns similare cu modelul de tranzacție de la HTTP, fiecare tranzacție fiind constituită de o cerere a unui client care invocă o metodă particulară sau funcție pe un server care ulterior răspunde prin cel puțin un răspuns. SIP reutilizează majoritatea câmpurilor de antet, reguli de codare și coduri de status ale HTTP furnizând un format bazat pe text.

Fiecare resursă a unei rețele SIP precum un user agent sau o căsuță vocală este identificată de un *Uniform Resource Identificier* (URI) precum în sintaxa utilizată de serviciile Web și e-mail, sintaxa standard folosită de SIP este ***sip:*** iar forma tipică a unui URI este : ***sip:username:***password@host***:port*** iar în cazul sistemelor criptate se folosește ***sips:*** astfel încât fiecare trecere printr-un nod să fie securizată cu TLS.

SIP funcționează în legătură cu alte protocoale însă rolul său este doar de semnalizare în cadrul sesiunii de comunicare. Clienții de SIP utilizează de obicei TCP sau UDP pe porturile 5060 pentru voce și 5061 pentru video sau comunicare securizată pentru a se conecta la serverele SIP sau sisteme destinație. Comunicarea audio și video se face prin intermediul protocolului RTP iar parametrii acestor streamuri sunt transmiși prin intermediul SDP în cadrul pachetului de SIP.

Motivația apariției protocolului SIP a fost de a semnaliza și pregăti comunicarea în cadrul comunicațiilor bazate pe IP ce poate suporta funcții de procesare și caracteristici prezente în rețelele de telefonie clasice(PSTN), protocolul în sine nu definește aceste funcții el concentrându-se pe creerea punții între clienți.

Protocolul SIP este un protocol client-server însă majoritatea sistemelor configurate pentru SIP pot acționa atât precum client cât și precum server. În general cel ce inițiază sesiunea este client iar cel ce recepționează convorbirea are rolul de server, caracteristicile SIP fiind implementate la capetele canalului contrar altor sisteme unde configurația se face pe rețea.

User Agent SIP (UA) este un nod logic din rețea folosit pentru a crea și primi mesaje SIP. Asftel controlând sesiunea SIP, agentul poate avea rolul de client (UAC) el trimițând cereri SIP dar și rolul de server (UAS) primind cereri și oferind răspunsuri, aceste roluri durând doar pe durata transacției SIP.

Un telefon SIP este un telefon IP care implementează funcțiile de user agent și server producând funcțiile de apel clasice ale unui telefon precum *formează, răspunde, respinge,reține sau transfer,* telefonul putând fi fizic sau software (*softphone*).

Mesajele SIP:
Cereri

REGISTER– folosit de un UA pentru a indica adresa IP curentă și lista de URL-uri de la care poate primi apeluri

INVITE – folosit pentru a stabili o sesiune între doi UA

ACK – confirmă siguranța schimbului de mesaje

CANCEL – încheie cererea aflată în așteptare

BYE – încheie sesiunea

OPTIONS – cere informații despre capabilitățile unui apelant fără a face legătura

Răspunsuri

1xx (PROVIZORIU): cererea primită și aflată în procesare

2xx (SUCCES): cererea a fost primită , înțeleasă și acceptată

3xx (REDIRECTARE): acțiuni viitoare trebuiesc să fie făcute de apelator pentru a completa cererea

4xx (EROARE CLIENT): cererea conține sintaxă proastă sau nu poate fi îndeplinită de server

5xx (EROARE SERVER): serverul nu a putut îndeplini o cerere aparent validă

6xx (EROARE GLOBALĂ): cererea nu poate fi îndeplinită de nici un server

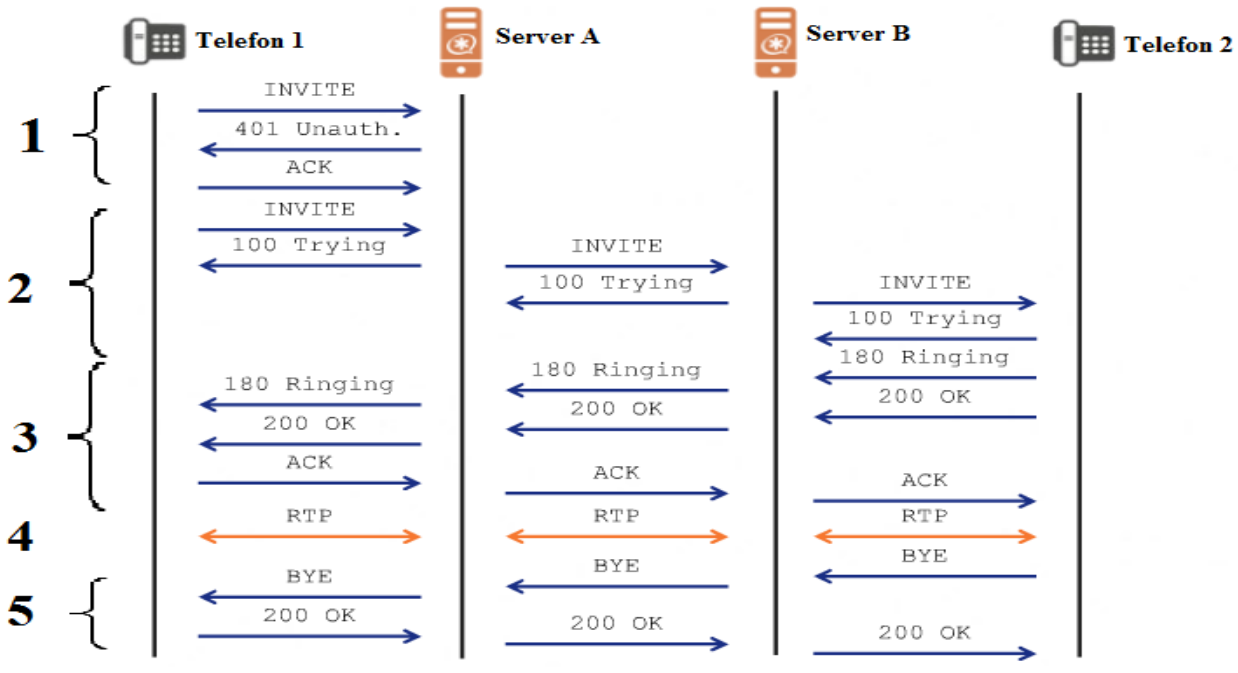

Figura 3.2. Exemplu comunicație SIP

În figura 3.2 este exemplificată o comunicație între două telefoane generice A și B prin două servere Asterisk, la pasul 1 telefonul A dorește să inițieze convorbirea cu telefonul B , pentru aceasta trimite un mesaj de *INVITE* , o cerere de legătură, serverului de Asterisk. În acest exemplu serverul A este configurat să solicite autentificarea astfel transmite înapoi telefonului mesajul *401 Unauthorized* ce ii sugerează telefonului că trebuie să furnizeze date de autentificare. Telefonul A răspunde cu un mesaj de confirmare și retransmite mesajul de *INVITE* modificat în pasul 2.

În pasul 2 telefonul A transmite noua cerere de legătură iar serverul răspunde cu mesajul *100 Trying*, dând astfel de înțeles că autentificarea a avut succes și încearcă legătura transmițând cererea de legătura mai departe către serverul B. Din motive de securitate când se face legătura între servere de asemenea serverul B poate cere serverului A să se autentifice. Serverul B verifică existența telefonului B în dialplan și solicită legătura, telefonul B răspunzând prin *100 Trying* și *180 Ringing* ce este propagat în pasul 3 până la telefonul A unde va începe să se audă tonul de apel.

În pasul 3 în cazul în care telefonul B răspunde se propagă mesajul *200 OK* până la telefonul A acesta răspunzând cu mesajul de confirmare. În momentul în care mesajul de confirmare ajunge înapoi la telefonul B înseamnă că s-a făcut legătura și se poate trece la pasul 4 de comunicații RTP.

În pasul 5 , atunci când unul din capete închide telefonul se propagă se propagă până în celălalt capăt mesajul *BYE* acest mesaj are rolul de ai îi spunde telefonului destinație să nu mai proceseze semnalul vocal și de a alerta utilizatorul că s-a încheiat conversația. În urma primirii mesajului *BYE* telefonul răspunde cu mesajul *200 OK* ce în momentul în care ajunge la destinație marchează încheierea conversației [8][9].

3.2. Asterisk

3.2.1. Prezentare centrală Asterisk

Asterisk este o implementare software a unui sistem de telefonie clasic, private branch exchange (PBX), creată în anul 1999 de compania Digium. Ca orice centrala telefonică, aceasta permite telefoanelor atașate să comunice între ele și să se conecteze la alte servicii de telefonie precum rețelele de telefonie fixă, *Public Switched Telephone Network* (PSTN) și servicii de Voice Over IP.

Asterisk este distribuită sub licență dual atât open-source (GPL)cât și cu licență de soft proprietar pentru a permite distribuția de componente de sistem nepublicate. Original centrala a fost construită să ruleze pe sisteme de operare Linux însă a fost importată și pe alte sisteme precum Mac OS X, Solaris, OpenBSD s.a.m.d, de asemenea centrala este suficient de compactă încât poate fi folosită și pe sisteme embedded.

Software-ul Asterisk include o mulțime de componente existente în sistemele de telecomunicații clasice precum voice mail, apel distribuit, *Interactive Voice Response* și distribuție automată a apelului; fiind un soft utilizatorii au posibilitatea să își creeze propriile funcționalități prin modificarea scripturilor de planificare a apelurilor (*dialplan scripts*) sau prin adăugarea de module scrise în limbajul C sau în

orice alt limbaj de programare cu funcționalități de comunicare prin fluxuri standard (*stdin și stdout*) sau socketuri TCP folosind *Asterisk Gateway Interface* (AGI).

Asterisk suportă o varietate mare de protocoale Voice Over IP incluzând SIP, centrala comportându-se atât precum un sistem de înrolare a telefoanelor cât și de punte între telefoanele software și PSTN, de asemenea Asterisk are propriul protocol de comunicare, *Inter-Asterisk eXchange*(IAX2).

Un server Asterisk nu este complet până când nu are telefoanele sau serviciul de telefonie configurat, acesta putând folosi tehnologii VoIP, telefoane clasice sau o combinație între ele.

- ***Conexiuni VoIP și telefoane*** – serverul de Asterisk se conectează la rețeaua locală folosind conexiunea Ethernet iar prin protocoale VoIP precum SIP comunică cu alte telefoane IP, provideri de servicii VoIP și alte servere VoIP; providerii de servicii VoIP oferă și posibilitatea comunicării cu rețelele PSTN.

-***Conexiuni PSTN și telefoane*** – conexiunea la rețeaua de telefonie clasică și telefoane se face utilizând o plăcuță de interfațare .

Plăcuțele de interfațare sunt dispozitive PCI ce sunt instalate pe placa de bază a serverului care au ca intrări și ieșiri pentru linia telefonică clasică sau telefoane. Plăcuțele sunt utilizate pentru servere reale(nu mașini virtuale), plăcuțele având rolul de a traduce semnalul telefonic clasic în formatul nativ al centralei. În cadrul centralei plăcuțele sunt configurate și controlate folosind Digium/Asterisk Hardware Device Interface (DAHDI)[10][11].

3.2.2. Configurări și standarde Asterisk

Pentru a putea utiliza centrala telefonică trebuiesc mai întâi create extensii pentru utilizatori din cadrul fișierului *extensions.conf* . Acest fișier este conține planul de apelare, centrul de control și execuție al tuturor operațiilor, aici se configurează tot comportamentul centralei telefonice.

Dialplanul este organizat în trei secțiuni:

- [**general**] – aici sunt setate diferite obțiuni generale precum păstrarea dialplanului sau eliberarea variabilelor.
- [**globals**]- aici sunt declarate variabilele globale dar care de cele mai multe ori sunt folosite drept constant, de exemplu se pot define numele canalelor: Callcenter => Telefon/1 sau se pot defini extensiile.
- [**contexts**]- aceasta este cea mai cuprinzătoare zonă, pot exista mai multe secțiuni de context fiecare fiind formată din diferite extensii. Atunci când centrala primește o conexiune de apel, fie că apelul provine din exterior fie de la o extensie internă acel apel aparține unui context, astfel contextele sunt folosite pentru a obliga centrala să acționeze diferit în funcție de unde provine apelul.

De exemplu în caz că linia este ocupată să redirecționeze apelul spre căsuța vocală sau în cazul în care apelul urmează să fie unul în afara țării să folosească canalul pentru VoIP în loc de cel normal [10].

Structura unui context este de forma exten => extensie, număr de prioritizare, comandă ; precum în exemplul:

```
[incoming]
exten => s,1,Answer()
exten => s,n,Playback(hello-world)
exten => s,n,Hangup()
```

unde *s* reprezintă extensia de start, orice apel din contextul *incoming* este răspuns automat și se rostește preînregistrarea *Hello-world* apoi se închide[10].

3.2.3. Decelare liniște-vorbire

Asterisk nu prezintă suport pentru voice activity detection datorită comunicării prin RTP , punctele terminale ce transmit informația audio sub formă de stream audio nu sunt nevoite să trimită pachete în perioadele de liniște, comunicarea RTP permințând transmisia necontinuă, operația de supresie a liniștii făcându-se printr-un marcaj.

Comunicarea prin RTP se face prin două canale, un canal de transmisie și unul de recepție, atunci când se dorește înregistrarea celor două canale există posibilitatea stocării canalelor separat, input și output sau multiplexat , alierea lor făcându-se după marcajul de transmisie.

În cadrul capitolului 4 soluția provizorie pentru a obține eșantioanele de vorbire, ale persoanei ce trebuie să fie înregistrată, a fost de a stoca, pe lângă canalul multiplexat, canalul ce aparține acelei persoane. Problema cu acest lucru este că fișierul stocat conține secvențe de vorbire urmate de perioade de liniște corespunzătoare cadrelor în care nu s-a făcut transmisia, adică cadrele în care vorbește cealaltă persoană. Pentru a rezolva această problemă, am aplicat algoritmul de decelare liniște – vorbire.

În figura 3.3. se observă faptul că zonele de liniște se caracterizează printr-un nivel de energie foarte mic și un număr de treceri prin zero mai mic decât cel al semnalului vocal, prezența trecerilor prin zero în zonele de liniște fiind datorate frecvenței rețelei ce afectează sistemul de captură a semnalului vocal. De menționat este faptul că sunetele sonore sunt caracterizate de o energie mare față de suntele insonore în timp ce sunetele insonore sunt caracterizate de un număr de treceri prin zero mai mare față de semnalele sonore.

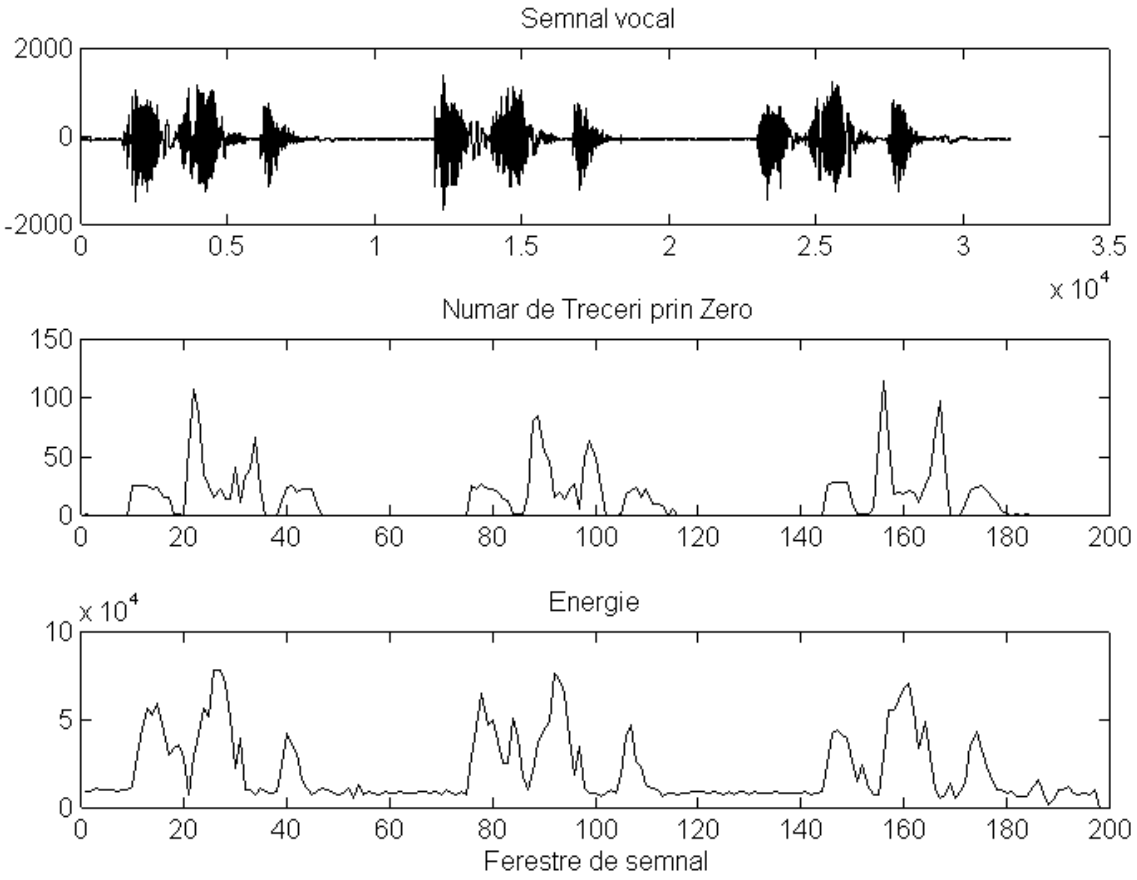

Figura 3.3. Energia și rata de treceri prin zero a unui semnal vocal [18]

Pentru a face decelarea liniște-vorbire se observă că informațiile ce sunt specifice porțiunilor de *liniște* sunt cele care dau detalii despre cum anume se diferențiază aceste zone de semnalul vocal util, astfel se se împarte semnalul vocal în cadre de lungime $20\,ms$ și se calculează energia și numărul de treceri prin zero pe fiecare cadru conform formulelor 3.1 respectiv 3.2.

$$E_m = \sum_{j=0}^{N-1} (x_m[j])^2 \tag{3.1}$$

$$rtz_m = \sum_{j=0}^{N-1} \frac{[1 - sgn(x_m[n]) \cdot sgn(x_m[n+1])]}{2} \tag{3.2}$$

unde *m* reprezintă numărul cadrului, *N* dimensiunea cadrului iar *sgn* reprezintă semnul eșantionului .

Se calculează valori ale energiei medii și ratei de treceri prin zero medii pentru primele *15* cadre de analiză ce se presupun a apaține zonei de liniște folosind formulele 3.3 și 3.4. De asemenea se calculează energia maximă din cadre mai puțin energia primelor *15* cadre.

$$EMedie = \frac{\sum_{k=1}^{15} E(k)}{15} \tag{3.3}$$

$$RtzMedie = \frac{\sum_{k=1}^{15} rtz(k)}{15} \tag{3.4}$$

$$EMax = \max\big(Etotal - E(1:15)\big) \tag{3.5}$$

Se calculează praguri pentru energie și rtz după cum urmează:

$$I_1 = C_1 * (Emax - EMedie) + EMedie \tag{3.6}$$

$$I_2 = C_2 * EMedie \tag{3.7}$$

$$ELowThreshold = \max\big(I_1, I_2\big) \tag{3.8}$$

$$EHighThreshold = C_3 * ELowThreshold \tag{3.9}$$

unde $C_1 = 0.03$, $C_2 = 5$, $C_3 = 3$ sunt valori determinate empiric.

$$RtzSigma = \sqrt{\frac{\sum_{k=1}^{15}(RtzMedie - rtz(k))^2}{15(15-1)}} \tag{3.10}$$

$$RtzThreshold = RtzMedie + 2 * RtzSigma \tag{3.11}$$

Cu ajutorul acestori praguri se pot lua decizii de start și de stop a cuvintelor precum se exemplifică în figurile 3.3 și 3.4.

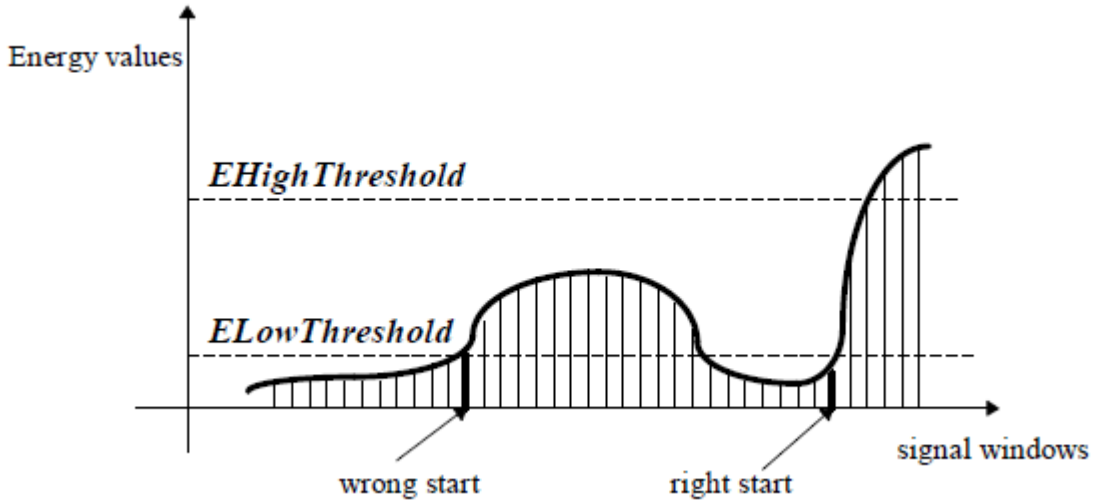

Figura 3.4. Detecția pe criterii energetice a începutului de cuvânt [18]

În figura 3.4 este reprezentat modul de luare a deciziei a începutului de cuvânt pe criteriu energetic, astfel începutul cuvântului este momentul în care energia unui cadru se ridică peste cele două praguri *ELowThreshold* și *EHighThreshold* reprezentând pragul de jos respectiv pragul de sus al energiei.

Se rafinează decizia de start a cuvântului prin folosirea ratei de treceri prin zero, începutul de cuvânt aflându-se la maxim *15* cadre sau la prima fereastră care scade sub nivelul stabilit *RtzThreshold* parcurgând semnalul vocal în sens invers pornind de la începutul de cuvânt ales energetic.

În mod similar, în figura 3.5 este reprezentat modul de luare a deciziei de sfârșit al cuvântului, pe criterii energetice sfârșitul cuvântului este acela în care timp de *15* cadre energia nu se ridică sub pragul de sus al energiei.

Se rafinează decizia de stop a cuvântului mergând înainte, de la poziția de stop aleasă energetic, maxim *15* ferestre, până la ultimul cadru ce depășește pragul ales al ratei de treceri prin zero [18].

Codul utilizat pentru această metodă este comentat în Anexa 1

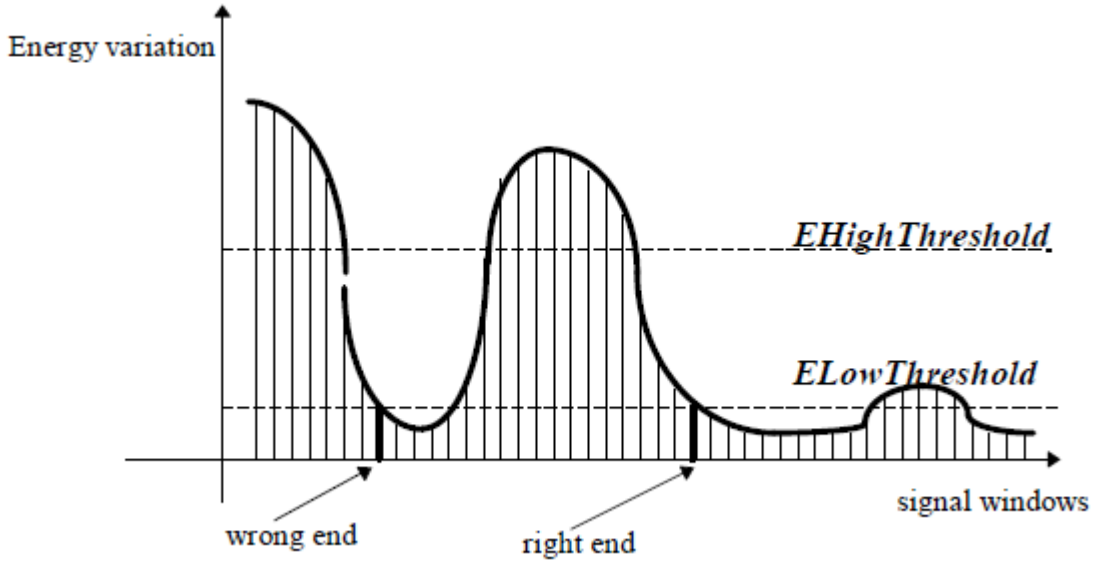

Figura 3.5. Detecția pe criterii energetice a sfârșitului de cuvânt [18]

Capitolul 4

Sistem de recunoaștere a vorbitorului în rețele de telecomomunicații

4.1. Alize

Alize este o platformă open-source folosită în recunoașterea vorbitorului, librăriile sale implementează metodele statistice folosite în domeniu precum modelarea cu mixturi Gaussiene dar și un set de metode dedicate recunoașterii vorbitorului precum Joint Factor Analysis, Support Vector Machine, modelarea cu i-vectori și Probabilistic Linear Discriminant Analysis.

Extragerea caracteristicilor se face folosind sistemele open-source Spro sau HTK, odată extrase acestea se normalizează, prin procesul menționat în subcapitolul 2.2.3, folosind funcția *NormFeat* , de asemenea se pot elimina ferestrele ce conțin un nivel scăzut de energie corespunzătoare liniștii sau zgomotului folosind funcția *EnergyDetector*.

Pentru extragerea caracteristicilor MFCC am folosit funcția *sfbcep* din cadrul utilitarului Spro cu parametrii:

sfbcep -F PCM16 -l 20 -d 10 -p 19 -e -D -A filename featurefilename

unde formatul înregistrării este precizat prin parametrul *–F PCM16*, codată PCM pe 16 biți, lungimea ferestrei de analiză este precizată prin parametrul *–l 20*, se aleg ferestre de 20 ms, *-d 10* reprezintă deplasarea ferestrei din 10 în 10 ms, *-p 19* reprezintă numărul de caracteristici extrase, s-au ales 19, *-e* reprezintă adăugarea energiei iar *–D –A* reprezintă concatenarea coeficienților delta și accelarația.

Funcția *TrainTarget* este folosită pentru a extrage modelul UBM folosind criteriul expectation maximization, aceasta primește ca intrare lista de fișiere de caracteristici extrase la pasul anterior. Se poate selecta numărul de mixturi Gaussiene prin parametrul `mixtureDistribCount` , pentru rapiditate am ales calcularea a 32 de mixturi . Un alt parametru este numărul de caracteristici din care să aleagă caracteristicile , deoarece în nici un experiment nu am avut un număr foarte mare de înregistrări am ales să aleagă toate caracteristicile. De asemena prin experimente am concluzionat că un număr de iterații mai mare de 5 nu imbunătățește performanța sistemului.

Funcția *TotalVariability* este folosită pentru a genera matricea *T* , aceasta primește ca intrare modelul UBM cu denumirea de *world.gmm* și fișierul ce conține numele fișierelor folosite în antrenarea modelului. Această funcție produce statisticile de oridinul 1 și 2 din cadrul analizei Baum-Welch pentru estimarea i-vectorilor, numărul de iterații folosit este 20.

Funcția *ivExtractor* este folosită pentru a extrage vectorii intermediari din matricea *T* corespunzători idurilor din fișierul de antrenare *ivExtractor.ndx*, ulterior cu ajutorul funcției *ivNorm* se normalizează vectorii conform metodei prezentate în subcapitolul 2.4.3 .

Analiza PLDA se face folosind funcția *PLDA* ce are rolul de a extrage matricile de voce și de canal *pldaEigenVoiceMatrix* și *pldaEigenChannelMatrix*. În cazul testării folosind funcția *testPLDA* se compară folosind analiză lineară vectorul intermediar cu aceste matrici.

Fiecare funcție necesită generarea unui fișier de configurare, mai exact, o listă a componentelor pe care trebuie să opereze. De exemplu pentru generarea modelului UBM se generează o listă numită *UBM.lst* care conține numele fișierelor de caracteristici iar pentru extragerea vectorilor intermediari se generează un fișier *ndx, ivExtractor.ndx* ce conține pe fiecare rând numele sub care trebuie salvat ivectorul și numele fișierului de caracteristic corespunzător. Reținem că ivectorii se generează presupunând că fiecare înregistrare aparține unui vorbitor diferit! Astfel ivectorii se salvează separat chiar dacă aparțin aceluiaș vorbitor de exemplu:

ivectorId1V1 featuresSpeaker1Utterance1
ivectorId1V2 featuresSpeaker1Utterance2
ivectorId1V3 featuresSpeaker1Utterance3

În cazul funcțiilor *PLDA* și *ivNorm* pe fiecare rând se trece numele ivectorilor aparținând unui vorbitor, de exemplu :

ivectorId1V1 ivectorId1V2 ivectorId1V3 etc.
ivectorId2V1 ivectorId2V2 ivectorId2V3 etc.

Antrenarea unui model se face prin generarea unui fișier *TrainModel.ndx* în care sunt trecute pe fiecare rând numele modelului urmat de numele ivectorilor corespunzători astfel:

SpeakerModel1 ivectorId1V1 ivectorId1V2 ivectorId1V3 etc.

iar în cazul testării se generează un fișier *ivTestTargetPLDA.ndx* ce conține pe fiecare linie numele ivectorului ce se dorește a fi testat urmat de numele modelelor cu care să testeze astfel:

ivectorIdTest1 SpeakerModel1 SpeakerModel2 SpeakerModel3 etc.

În urma testării se generază un fișier ce conține scorurile obținute prin compararea unui ivector cu toate modelele.

4.2. Baze folosite în testarea sistemului

Pentru evaluarea sistemului s-au folosit 4 baze, o bază folosită în recunoașterea de vorbitor formată 24 de vorbitori, fiecare având 60 de secvențe de vorbire în care sunt rostite enunțuri scurte , înregistrările nedepășind 10 secunde de vorbire, înregistrările au fost făcute în condiții de liniște și stocate în formatul telefonic *G712* codate PCM. A doua bază de înregistrări este o bază formată din 83 de vorbitori împărțiți pe sexe, fiecare vorbitor rostind numere în condiții de zgomot ambiental(se aud ventilatoare, neoane) lungimea înregistrărilor fiind între 10 și 30 de secunde. Celalte două baze au fost generate pentru a simula convorbirile reale și pentru a studia efectul mesajului vorbit cum se detaliază în cele ce urmează.

Pentru a simula convorbiri reale, a fost instalată centrala telefonică Asterisk conform specficiațiilor oferite de Digium [11] și s-a modificat dialplanul astfel încât să se stocheze, pe lângă canalul mixat, doar canalul corespuzător vorbitorului ce se dorește a fi înrolat folosind funcția *Monitor* în cadrul contextului de *[record]* .

Funcția *Monitor* este comanda ce inițializează procesul de înregistrare a canalului, intrarea și ieșirea canalului este stocată în fișiere diferite însă acestea pot fi combinate folosind parametrul *m* . Pentru a stoca doar canalul dorit se poate folosi parametrul *i* atunci când se dorește salvarea canalului persoanei apelate sau parametru *o* când se dorește salvarea canalului persoanei care apelează. Înregistrările sunt stocate în format WAV codate PCM pe 16 biți, de asemenea se poate specifica numele înregistrării, în mod standard Asterisk stochează înregistrările astfel :

Exten-apelant-apelat-zi/luna/an-idapel-o/i/m.wav

Pentru generarea bazei din producție a fost introdus scriptul modificat în dialplanul centralei telefonice din producție și s-au stocat înregistrările doar de la anumiți operatori ce nu procesează date confidențiale, dezavantajul acestei metode este faptul că nu toate înregistrările conțin informații utile. În majoritatea cazurilor înregistrărilor nu au fost inițiate cu succes sau conțin mesajele de căsuță vocală ale apelaților. De asemenea funcția *Monitor* consideră inițierea convorbirii din momentul în care se răspunde cu tonul de apel, motiv pentru care baza a trebuit să fie procesată prin eliminarea segmentelor de început și a înregistrărilor inutilizabile, din 435 de înregistrări au rămas doar 307 din care 32 aparțin aceluiași vorbitor, acestea au fost folosite în testare.

Baza de 307 de vorbitori a fost preprocesată folosind scriptul de decelare liniște/vorbire, s-au întâmpinat probleme la înregistările ce aveau mesaj de întâmpinare a apelului unde vocea căsuței vocale automate a fost de asemenea segmentată astfel aceste înregistrări au fost eliminate, baza finală conținând 277 de vorbitori.

Pentru generarea bazei de test folosită în prezentarea prototipului s-au înrolat 20 de vorbitori, fiecare vorbitor a fost rugat să rostească un cod generat aleator de lungimea unui CNP de două ori spre a fi folosit în antrenare și un al doilea cod din cele generate, corespunzătoare celorlalți vorbitori pentru a fi folosit în testare. Scopul acestui experiment este de a vedea dacă este recunoscut vorbitorul în loc să fie recunoscut mesajul rostit.

4.3. Experimente

În primă fază s-a testat funcționalitatea exemplului propus de Alize. Pentru testare a fost folosită baza de înregistrări cu 83 de vorbitori; fiecare vorbitor are câte 5 înregistrări, primele patru înregistrări au fost folosite pentru antrenare iar a cincea a fost folosită în testare. Cu alte cuvinte s-au folosit 332 fișiere pentru antrenare și 83 fișiere pentru testare. Pentru a antrena s-au generat două scripturi numite *enrollment.sh* și *identification.sh*, prezente în Anexa 2 respectiv Anexa 3.

Pentru antrenare s-au generat fișierele *UBM.lst, ivExtractor.ndx, ivNorm.ndx, Plda.ndx, totalvariability.ndx și TrainModel.ndx* precum se poate vedea în fișierul *enrollment.sh* iar la testare s-au generat doar fișierele *ivExtractor.ndx* și *ivTestPlda.ndx*. Pentru generarea acestui din urmă fișier s-au folosit datele din fișierul de antrenare a modelelor. În urma testării s-au obținut scoruri precum în figura 4.1. în primă fază s-a ales valoarea maximă a scorului pentru toate fișierele testate.

În figura 4.1 este exemplificat cazul testării unei înregistrări aparținând primului vorbitor. Se observă că scorul maxim este dat în dreptul primului model, celalalte fiind negative mai puțin unul.

Pentru a pune condiția să se aleagă scorul maxim s-a generat scriptul *extractResults.sh* și după cum se observă în tabelul 4.1 s-a obținut o precizie de 84.34% asta doar în cazul în care considerăm primul maxim, la o inspecție mai apropiată a scorurilor s-a observat că scorul maxim obținut de un impostor nu diferă foarte mult de cel al persoanei reale, așa că a urmat să se verifice dacă vorbitorul real se află în primii doi vorbitori detectați. Precizia a crescut cu aproximativ 5% apoi în cazul în care s-a ales și al treilea maxim precizia a cresut la 95%.

Caz testare	Test pozitiv	Test negativ	Precizie(%)
Primul maxim	70	13	84.34
Al doilea maxim	74	9	89.16
Al treilea maxim	79	5	95.18

Tabelul 4.1. Testare bază 83 vorbitori

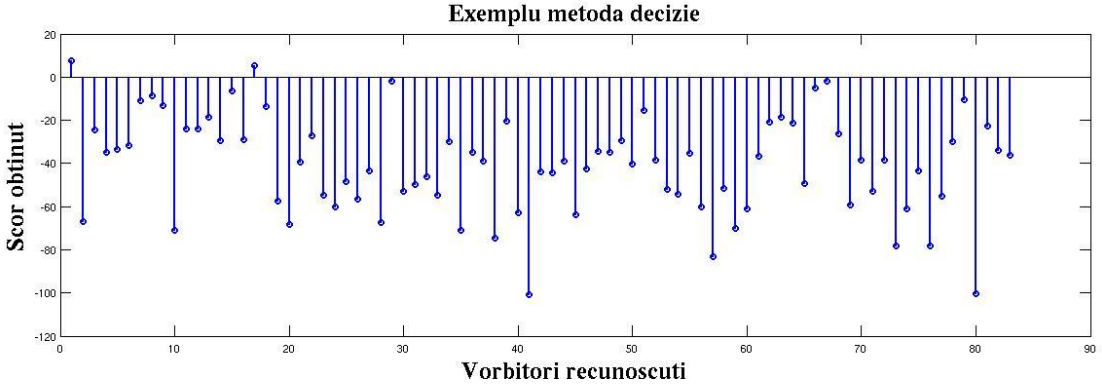

Figura 4.1. Exemplu scoruri obținute prin testarea unei singure înregistrări

Acest test nu este unul concludent, testarea unui sistem biometric trebuie să fie făcută folosind ratele de falsă acceptare și falsă respingere. Astfel în figura 4.2 se observă graficul dintre FAR și FRR, acest grafic a fost obținut prin calcularea valorilor FAR și FRR la diferite valori ale pragului de decizie. Deși majoritatea scorurilor vorbitorilor reali sunt pozitive s-a pornit incrementarea pragului de la -20(se observă în figura 4.1 că scorurile impostorilor oscilează în jurul valorii -20) până la valoarea 10 (valoarea maximă a scorurilor obținute este 9.34) cu un pas de 0.1.

Se observă ca valoare ratei de egalitate dintre FAR și FRR este la valoarea pragului de 0,024. S-a ales acest prag pentru a vedea cum arată matricea de confuzie exemplificată în figura 4.3. Cu noul prag de decizie de 0,4 s-a parcurs fișierul cu scoruri și s-au stocat doar valorile mai mari decât acesta într-o matrice astfel încât pe coloane să fie modelul vorbitorului iar pe linii să fie înregistrarea testată, rezultatul optim fiind când scorul maxim apare pe diagonala principală.

Pentru pragul de 0,024 s- au obținut o valoare a autentificărilor valide de 80, o precizie de 96,3% iar dacă testăm dacă maximul se află pe diagonala principală, întradevăr se obțin 70 de înregistrări valide precum în tabelul 4.1.

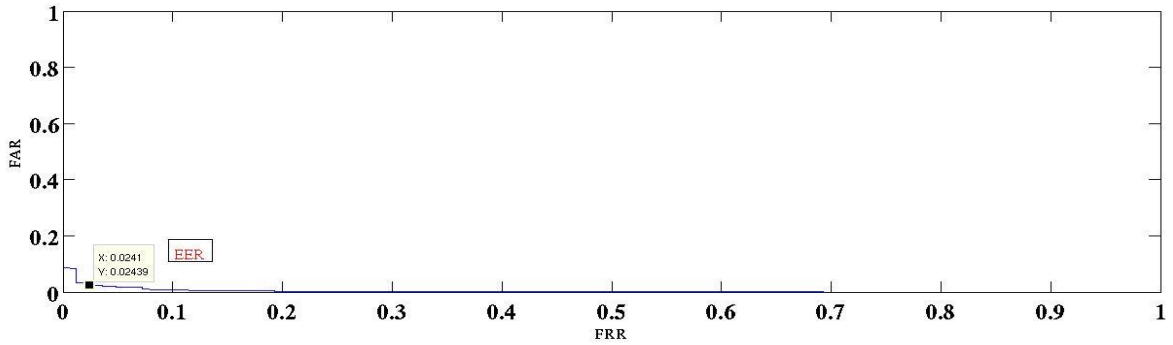

Figura 4.2. Curba ROC - baza de 83 vorbitori

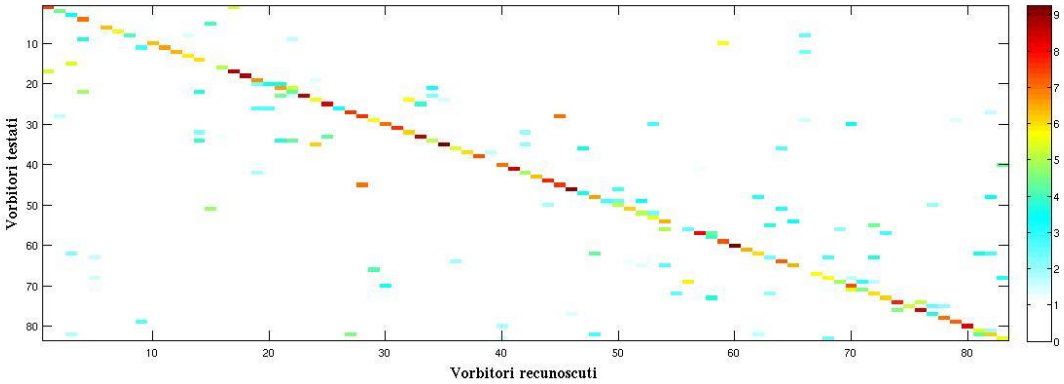

Figura 4.3. Matricea de confuzie 83 – baza vorbitori

Se observă în figura 4.3 că pe diagonla principală se află culorile cele mai accentuate. Procesul de verificare poate fi îmbunătățit prin stabilirea unei marje de eroare pentru fiecare vorbitor, calculând diferența între scorul de autentificare adevărată a vorbitorului și scorul obținut de următoarea persoană autentificată negativ.

Trebuie menționat faptul că atunci când selectăm scorurile maxime obținute are loc un proces de identificare iar atunci când se analizează sistemul biometric analiza se face pentru procesul de verificare.

În cazul bazei TSP se repetă experimentul anterior în care se aleg scorurile maxime de la fiecare înregistrare testată și se obține un procent de 59.16% înregistrări identificate corect, un procent foarte slab, luând în considerație faptul că s-au antrenat 29 de înregistrări și sau testat cu restul de 29 precum în tabelul 4.2.

	Antrenare 29 înregistrări	Primul maxim	Al doilea maxim	Al treilea maxim
Precizie(%)	Test 29	59	81	90
	Antrenare 29 voci feminine			
	Test 29 voci feminine	76	89	95
	Train 29 voci masculine			
	Test 29 voci masculine	87	98	99.6
	Media	81.5	93.5	97.3

Tabelul 4.2. – Antrenare și testare cu 29 de înregistrări mixte și pe genuri

	Antrenare 5 înregistrări	Primul maxim	Al doilea maxim	Al treilea maxim
Precizie(%)	Test 5	42	72	81
	Test 15	44	67	79
	Test 25	45	72	78
	Test 35	44	66	77
	Test 45	43	64	76
	Test 53	44	64	75

Tabelul 4.3. – Antrenare cu 5 înregistrări și testare cu 5-53 de înregistrări

Pentru a identifica problema s-au antrenat 5 înregistrări și s-a testat cu alte 5 înregistări precum în tabelul 4.3. Se observă în tabel că precizia sistemului este în continuare mică, mai mică decât cazul antrenării și testării cu 29 de antrenări. De asemenea se observă că dacă creștem numărul de înregistrări testate scorul obținut nu se modifică foarte mult ceea ce sugerează o robustețe la variabilitatea vorbirii.

În cazul evaluării sistemului biometric, se observă în figura 4.4 că pragul de egalitate între FAR și FRR este la *0.1*. Acest prag apare la valoarea pragului de decizie de -0.4. Se rulează procesul de verificare folosind acest prag și se obține matricea de confuzie din figura 4.5.

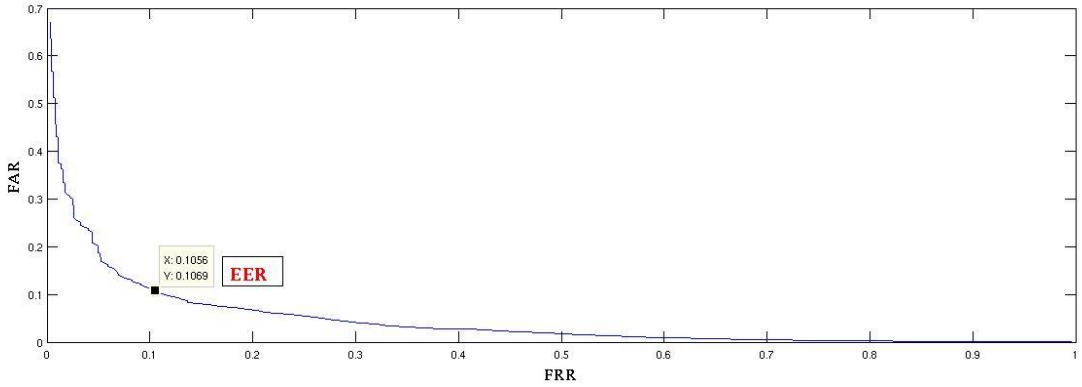

Figura 4.4 Curba ROC – baza TSP

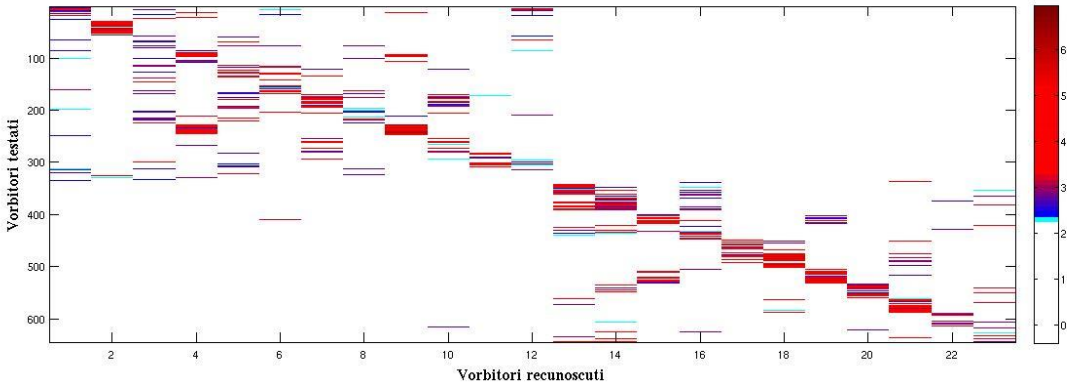

Figura 4.5. Matricea de confuzie – baza TSP

În figura 4.5 atrage atenția faptul că atunci când s-au testat înregistrări cu voci femine nu au fost detectate voci masculine și invers. Acest lucru sugerează robustețea la diferența dintre genuri însă în tabelul 4.2 se observă că prin antrenare separată pentru fiecare gen se obțin rezultate mult mai bune, media celor două sisteme fiind de 97,3 atunci când este luat în considerare și al treilea maxim.

	Antrenare 5 înregistrări voci masculine /voci feminine	Primul maxim	Al doilea maxim	Al treilea maxim
Precizie(%)	Test 5 voci masculine	56	91	95
	Test 5 voci feminine	50	85	92
	Media	53	88	93.5

Tabelul 4.4. – Antrenare cu 5 înregistrări și testare cu 5 diferențiate pe genuri

În tabelul 4.4 se observă că precizia când antrenăm cu mai puține înregistrări diferențiate pe genuri este mai bună decât în cazul antrenării mixte însă cu toate acestea pentru simplitudine toate experimentele din acest punct au fost făcute antrenând vocile mixt.

Baza cu 20 de vorbitori conține 13 voci feminine și 7 voci masculine. S-a repetat testarea considerând primul, al doilea și al treilea maxim din scorurile obținute precum în tabelul 4.5. De asemenea s-a refăcut și testarea diferențiată pe genuri.

	Voci	Mixte	Masculine	Feminine
Precizie(%)	Primul maxim	85	85,7	58
	Al doilea maxim	90	100	76
	Al treilea maxim	95	100	100

Tabelul 4.5. - Testarea cu voci mixte, masculine și feminine ținând cont de scorurile maxime

Pentru calculul curbei ROC s-au folosit înregistrările mixte. Se observă că valoarea de egalitate între FAR și FRR este de *0.1*, conform cu figura 4.6, valoarea pragului de decizie în această poziție fiind *-26*.

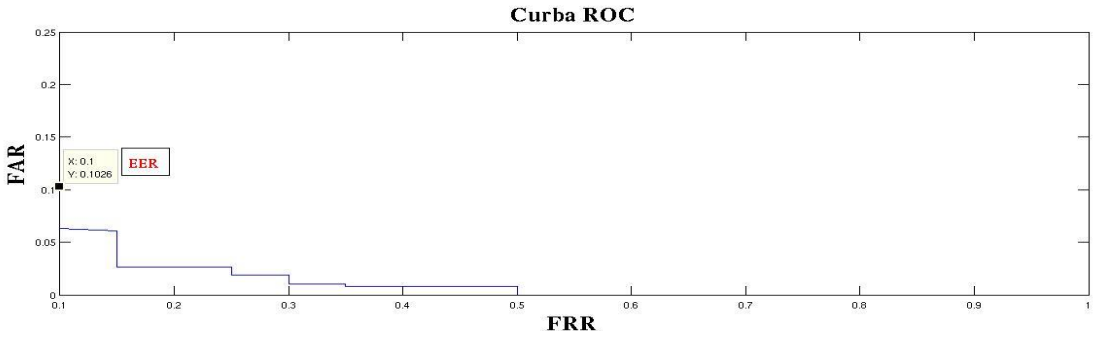

Figura 4.6. Curba ROC – baza 20 de vorbitori

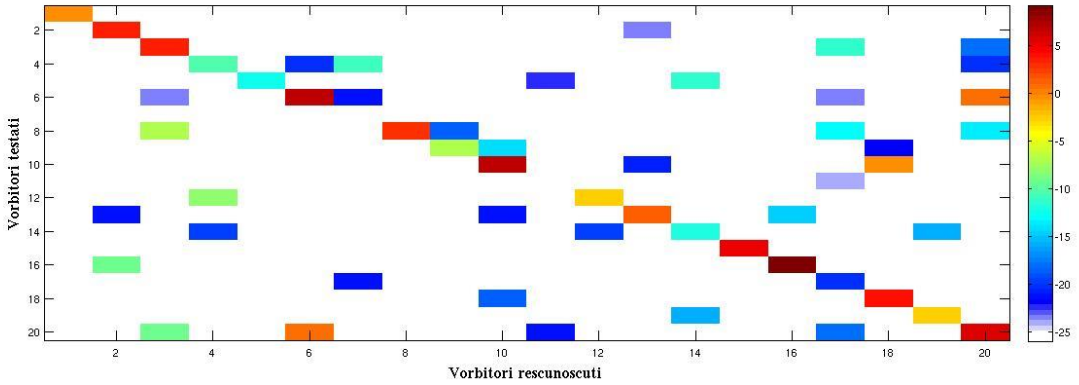

Figura 4.7. Matricea de confuzie – baza 20 vorbitori

În figura 4.7 se observă faptul că scorurile maxime se află pe diagonala principală cu ecepția vorbitorului 7 și 11. Vorbitorul 7 nu a fost detectat deloc iar vorbitorul 11 a fost confudat cu vorbitorul 17. Precizia acestui sistem de verificare, punând condiția ca scorurile maxime se află pe diagonala principală este de 90%. De asemenea se observă posibilele autentificări false.

Scopul acestui bazei este de a verifica dacă este recunoscut vorbitorul mai bine decât enunțul pronunțat, astfel s-au înregistrat pentru fiecare vorbitor două rostiri ale unui cod de lungimea unui CNP și o a treia rostire a unui cod aparținând altui vorbitor precum în tabelul 4.6.

Id vorbitor	CNP utilizator real	CNP impostor	Id utilizator real	Detectie
1	3176261195912	2540762438438	2	0
2	2540762438438	3038182215765	7	0
3	5869560700841	3176261195912	14	0
4	3228254719637	3176261195912	14	0
5	3544214696157	3228254719637	4	0
6	7766746673732	3391466907691	20	1
7	3038182215765	3544214696157	5	0
8	9944512040345	3824375631287	13	0
9	4183978710789	4143338546156	16	0
10	4226211935747	4183978710789	9	0
11	9456627607811	4224489043932	12	0
12	4224489043932	4226211935747	10	0
13	3824375631287	4588367233518	15	0
14	3176261195912	5087522844318	19	1
15	4588367233518	5869560700841	3	0
16	4143338546156	7456476785708	18	0
17	8376516768708	7766746673732	6	0
18	7456476785708	8376516768708	17	0
19	5087522844318	9456627607811	11	0
20	3391466907691	9944512040345	8	0

Tabelul 4.6. – Generea bazei de 20 de vorbitori și rezultatul testării

Pentru a genera câmpul *Detecție* din tabelul 4.6 s-a verificat conform cu figura 4.6 dacă vorbitorul a cărui cod a fost rostit a fost confundat cu cel care a rostit codul. Apar doar două astfel de situații vorbitorul 6 a rostit codul vorbitorului 20 și s-a obținut un scor ce trece de pragul de *-26*. Însă scorul obținut prin compararea înregistrării 6 cu vorbitorul 20 este apropiat de valoarea 0 în timp ce scorul comparării înregistrării 6 cu vorbitorul 6 este de aproximativ 5. Similar se întâmplă și în cel de-al doilea caz, lucru ce sugerează că sistemul este robust la rostirea unei fraze de autentificare de către un impostor.

Baza cu 245 de vorbitori este formată din înregistrări de durată variabilă, între 25 de secunde și 10 minute, pentru antrenare s-au segmentat înregistrările în secvențe de 15 secunde și s-au folosit doar două secvențe pentru a cuprinde toată populația de vorbitori. De asemenea înregistrările folosite în testare au fost segmentate în secvențe de 15 secunde și s-au folosit tot câte 2 înregistrări.

	Vorbitori	Primul maxim	Al doilea maxim	Al treilea maxim
Precizia(%)	245	65	80	83

Tabelul 4.7.- Precizia bazei de 245 vorbitori testând 64 de înregistrări

În tabelul 4.7 se observă rezultatul testării a 64 de înregistrări, câte două de la fiecare vorbitor testat. Scorul obținut este similar cu cel al bazei TSP cu observația că s-au antrenat doar 30 de secunde de vorbire precând în cazul bazei TSP s-au antrenat aproximativ 50.

În cazul evaluarii sistemului din punctul de vedere al operației de verificare s-a extras curba ROC precum în figura 4.8 și se observă că punctul de egalitate între FAR și FRR se află la poziția *0,03* corespunzătoare unui prag de decizie de *2.8* . Pentru pragul ales de precizia sistemului în condițiile în care maximul se află pe diagonala principală este de *96.8%* , adică 31 de vorbitori au fost fost autentificați corect.

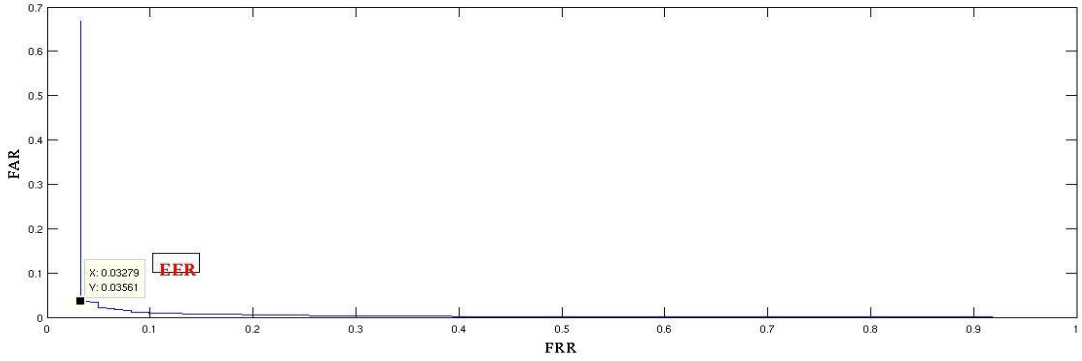

Figura 4.8. Curba ROC – baza 245 vorbitori

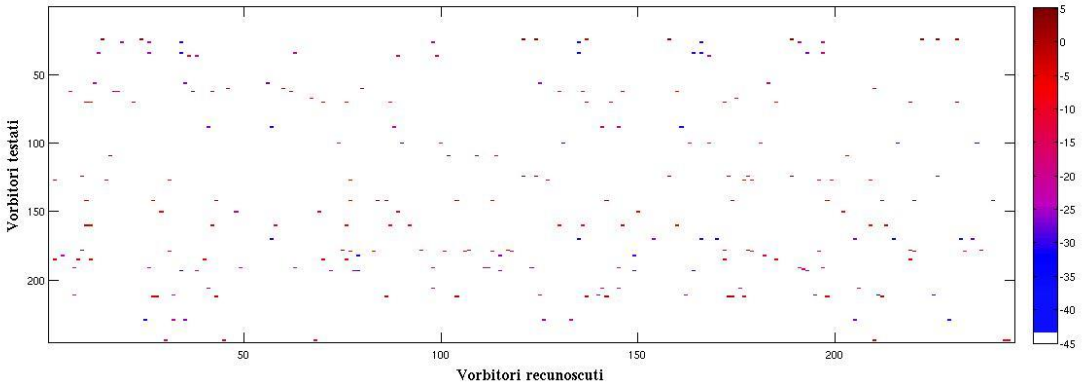

Figura 4.9. Matricea de confuzie – baza 245 vorbitori

În figura 4.9 este reprezentată matricea de confuzie. De remarcat este faptul că există un număr foarte mare de fals pozitive însă valorile maxime în continuare se află pe diagonala principală.

	EER	Prag decizie
Baza 83 vorbitori	0.024	0.4
Baza 23 vorbitori	0.1	-0.4
Baza 20 vorbitori	0.1	-26
Baza 245 vorbitori	0.03	2.8

Tabelul 4.8. – Comparație valori EER ale bazelor

În tabelul 4.8 sunt prezentate punctele de egalitate între ratele de falsă respingere și falsă acceptare. Se observă că din valorile EER calculând invers formula 4.42 pentru baza de 83 de vorbitori pot avea loc 2 autentificări false. De asemenea pentru baze cu 20 și 23 de vorbitori pot avea loc 2 autentificări false. Pentru baza cu 245 de vorbitori pot fi autentificate fals 7 înregistrări.

		83 vorbitori	23 vorbitori	20 vorbitori	245 vorbitori
Precizia(%)	Primul maxim	84.3	59	85	65
	Al doilea maxim	89.16	76	90	80
	Al treilea maxim	95.18	87	95	83
	Verificare	96.3	97.6	90	96.8

Tabelul 4.9. – Comparația preciziei între baze

În tabelul 4.9 sunt prezentate rezultatele testării tuturor celor 4 baze. Trebuie menționat faptul că valorile din contextul primul maxim, al doilea maxim , al treilea maxim se referă la cazul de identificare. S-a testat o înregistrare cu modelele tuturor vorbitorilor și s-a verificat dacă scorul vorbitorului real se află în primele trei scoruri detectate.

Câmpul *Verificare* se referă la procesul de verificare. Pentru a face verificarea s-a comparat înregistrarea de test cu modelele tuturor vorbitorilor și s-a calculat rezultatul obținut prin modificarea unui prag de decizie. S-a considerat pragul optim valoarea pentru care rata de falsă acceptare este egală cu rata de fasă respingere. Valorile obținute pentru pragul optim sunt trecute în dreptul câmpului *Verificare.*

4.4. Aplicație sistem recunoaștere de vorbitor

Aplicația de recunoaștere de vorbitor poate fi implementată sub forma de aplicație web. În urma unei convorbiri, se poate verifica dacă vorbitorul a fost deja înrolat sau dacă este nevoie să fie înrolat.

În cazul în care un vorbitor este existent într-o bază de date se poate lua decizia de a adăuga un nou model sau de a verifica dacă vorbitorul este cel care pretinde a fi.

În figura 4.10 se observă o astfel de aplicație, interfață grafică, generată în matlab folosind modul *Graphical User Interface*. Codul este generat automat iar scripturile de antrenare și testare sunt rulate direct din linia de comandă folosind comanda *unix.*

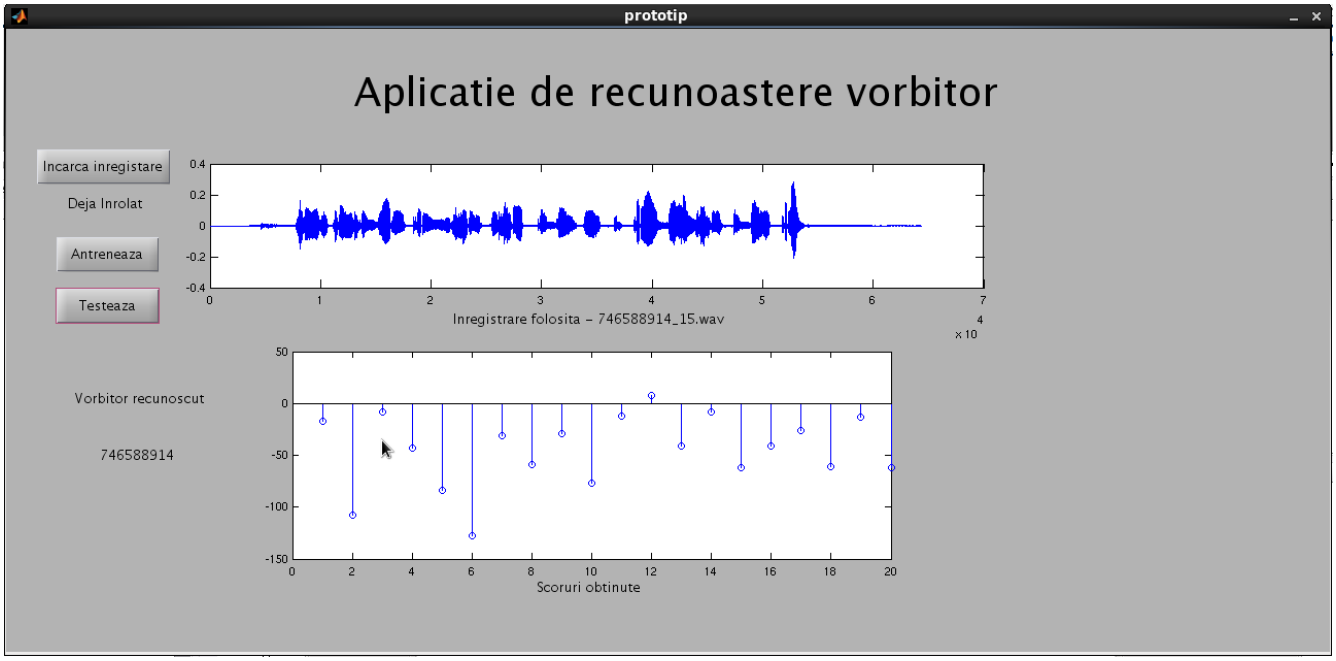

Figura 4.10. – Interfața grafică a sistemului de recunoaștere vorbitor – captură Matlab

Capitolul 5

Concluzii

În acest capitol sunt comentate pe scurt metodele folosite și rezultatele obținute în urma testării sistemului de recunoaștere de vorbitor în diferite scenarii.

Sistemul de recunoaștere de vorbitor este implementat folosind platforma open-source numită Alize. Parametrii de vorbire folosiți sunt coeficienții MFCC datorită ușurinței cu care se extrag și robustețea la zgomot. De asemenea prin derivarea coeficienților se oferă informații despre stilul și durata vorbirii.

Pentru a genera modelul vorbitorului se folosesc vectorii intermediari ce sunt normalizați prin analiză liniară iar efectul canalului de comunicații este normalizat folosind analiza în interiorul clasei.

Vectorii intermediari sunt vectorii propii ai matricei de variabilitate totală. Matricea de variabilitate este generată prin analiză statistică a modelul universal.

Se extrage câte un vector pentru fiecare rostire, modelul generându-se prin analizarea vectorilor unui vorbitor.

Metoda este robustă atât la variabilitatea vorbitorului cât și la variabilitatea canalului de comunicații.

S-au testat patru baze de înregistrări în diferite contexte.

S-a testat robustețea sistemului de recunoaștere la variabilitatea inter vorbitori pe o bază cu 83 de vorbitori. De asemenea s-a luat în considerare faptul că înregistrările au o lungime de 30 de secunde. S-au folosit 4 înregistrări în antrenare și 1 în testare obținându-se o precizie de ***96.3%***. Acest lucru sugerează robustețea din premiză.

De asemenea s-a testat robustețea sistemului de recunoaștere la genul vorbitorului, pe o bază cu 23 de vorbitori. Inițial s-a antrenat sistemul cu voci mixte și s-a obținut o precizie de 81%. În cazul antrenării cu 12 voci masculine și 11 voci masculine. S-a obținut o precizie de ***95%*** respectiv ***92%***. Trebuie reamintit faptul că în cazul antrenării mixte nu s-au detectat vorbitori masculini când s-a testat cu vorbitori femini și viceversa. Astfel sistemul este robust la genul vorbitorului.

Al treilea test a fost pe o bază de 20 de vorbitori generată cu scopul de a testa robustețea la mesajul vorbit în sensul că este recunoscut mesajul rostit în loc să fie vorbitorul.

Vorbitorii testați a căror scoruri se află peste pragul impus nu au corespondenți în dreptul vorbitorilor a căror mesaj de autentificare a fost rostit. Sistemul este robust la mesajul rostit.

Testul final este pe o bază de 245 de vorbitori. Înregistrările au fost extrase din convorbiri reale din cadrul callcenterului și au o lungime maximă de 30 secunde precum în primul test. Acest test este cel mai concludent. Sistemul este robust la variabilitatea inter vorbitori, s-a obținut o precizie de ***96.8%***. Sistemul este robust la variabilitatea canalului de comunicații. Clienții au sunat din locații diferite, folosind atât telefoane fixe cât și mobile.

Eșantioanele de vorbire ale vorbitorilor s-au obținut prin modificarea dialplanului din centrala de Asterisk pentru a stoca canalul persoanei apelate sau care apelează. Înregistrările se salvează în format WAV codate PCM pe 16 biți. Problema întâmpinată a fost că înregistrările conțin mesaje de căsuță vocală sau tonuri de așteptare. De asemenea înregistrările se salvează cu pauze în vorbire corespunzătoare timpilor în care vorbește operatorul de callcenter. Ambele probleme au fost rezolvate prin aplicarea unui cod de decelare liniște-vorbire studiat în cadrul laboratorului de *Tehnologia Vorbirii,Recunoașterea Vorbitorului*.

Sistemul de recunoaștere de vorbitor oferă o precizie acceptabilă și este robust la variabilitatea vorbitorului, a canalului de comunicații și a mesajelor rostite.

În viitor trebuiesc automatizate toate procesele și studiat modul în care se poate face integrarea sistemului pe un server proiectat pentru procesare paralelă. Sistemul de recunoaștere trebuie să ruleze în mai multe sesiuni simultane în cadrul callcenterului.

Bibliografie

[1] Lisa Myers, **"*An Exploration of Voice Biometrics",*** 2004
[2] Matt Warman, **"*Say goodbye to the pin: voice recognition takes over at Barclays Weath"***, 2013
[3] Adam Vrankulj, ***„My Verified ID partners with HealthCare Select for biometric verification"***, 2014
[4] Zhu,Xuan,***"Improved speaker diarization using speaker identification"***
[5] Sadaoki Furui,***"50 years of progress in speech and speaker recognition"***
[6] Roberto Togneri, Daniel Pullella, ***"An Overview of Speaker Identification: Accuracy and Robustness Issues"***
[7] Howard Lei, ***"Joint Factor Analysis and i-vector tutorial"***
[8]Najim Dehak, Stephen Shum, ***"Low-dimensional speech representation based on Factor Analysis and its application",*** Spoken Language System Group MIT
[9] SIP RFC http://www.ietf.org/rfc/rfc3261.txt
[10] Tutorial Asterisk http://www.digium.com/en/training/asterisk/asterisk-essentials-training
[11] www.digium.com
[12] http://www.bromba.com/faq/biofaqe.html
[13] S. Pruzansky, "***Pattern-matching procedure for automatic talker recognition***,", pp. 354-358, 1963
[14] S. Furui***, "An analysis of long-term variation of feature parameters of speech and its application to talker recognition" ,*** Electronics and Communications in Japan ,57-A, pp. 34-41, 1974
[15] S. Furui, "***Cepstral analysis technique for automatic speaker verification***," IEEE Trans. Acoustics, Speech,
Signal Processing, ASSP-29, pp. 254-272, 1981
[16] T. Matsui and S. Furui, "***Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs***," Proc. ICSLP, pp. II-157-160, 1992
[17] T. Matsui and S. Furui, "***Concatenated phoneme models for text-variable speaker recognition***," Proc. ICASSP,4 pp. II-391-394, 1993.
[18] Tema de casă – laborator TVRV
[19] Dragoș Burileanu, note de curs *Tehnologii Biometrice, Recunoașterea Semnăturii Dinamice*

Anexa 1
Cod decelare liniște-vorbire

```
function [index_start,index_stop]=delimitare_semnal_vocal(root,filename,dest)
mkdir([dest filename]);
%declarare de variabile
C1=0.03;
C2=5;
C3=3;
CL=15;
p=0;o=0;u=0;
start=0;stop=0;
realstart=0;realstop=0;
starts=[];
stops=[];

[inreg,fs]=wavread([root filename]);
%plot(inreg)
len=length(inreg);
 %figure(),plot(0:1/fs:(len-1)/fs,inreg)
dur=160;  %lungimea ferestrei de 20ms
[energy,ntz]=en_ntz(inreg,dur);
[energymed,ntzmed]=en_ntz(inreg(1:CL*dur),dur);

%calculul lui EMedium pe 15 ferestre de la inceputul semnalului
energymed=sum(energymed)/CL;

%calculul lui NTZMedium pe 15 ferestre de la inceputul semnalului
ntzmed=sum(ntzmed)/CL;

%calculul lui EMaximum
Emax=max(energy(CL:end));

%calculul pragurilor ,,
I1=C1*(Emax-energymed)+energymed;
I2=C2*energymed;
%ELowThreshold
Elowth=max(I1,I2);
%EHighThreshold
Ehighth=C3*Elowth;

%NTZsigma
for i=1:CL
    ntzdif=ntzmed-ntz(i);
    ntzpow(i)=ntzdif^2;
end
Ntzsigma=sqrt(sum(ntzpow)/(CL*(CL-1)));
%NTZThreshold
Ntzth=ntzmed+2*Ntzsigma;

lenen=length(energy);
k=0;
while k<lenen
    k=k+1;
    %decizia de inceput a cuvantului pe criterii energetice
```

```
    %parcurgem inregistrarea pana gasim valori mai mari decat pragul de jos
    if energy(k)>Elowth
        while (realstart==0 & k<lenen)
            %continuam sa parcurgem pana gasim valori ale pragului de sus
              k=k+1;
            if energy(k)>Ehighth
                while (realstart==0 & k<lenen)
                    k=k-1;
                    %dupa ce gasim pragul de sus parcurgem invers
                    %inregistrarea pana gasim prima valoare sub pragul de
                    %jos, valoarea de start
                    if energy(k)<Elowth
                        if energy(k-1)>Elowth k=k-1;
                        else
                        realstart=k;
                        o=o+1;
                        %formam vector ai timpilor de start
                        starts(:,o)=realstart;
                        end
                    end
                end
            end
        end
    end
     %decizia de sfarsit a cuvantului pe criterii energetice
     %parcurgem inregistrarea din pozitia de start a cuvantului
   while (realstart~=0 & k<lenen)
       k=k+1;
       %parcurgem inregistrarea cat timp nu scade sub pragul de jos
       if energy(k)<Elowth stop=k;
       end
      %cand scade sub pozitia de minim cautam daca timp de 15 ferestre nu
      %creste peste pragul de jos
       while (stop~=0 & k<lenen)
           k=k+1;
           p=p+1;
           if k>lenen-1
               realstop=stop;
               stops(:,o)=realstop;
               realstart=0;
           end
           %daca semnalul creste peste pragul de jos reincepem cautarea
           if energy(k)>Elowth
               p=0;stop=0;
           end
           if p==CL realstop=stop;
               stops(:,o)=realstop;
               realstart=0;
           end
       end
   end
end

len2=length(stops);
fakestarts=starts;
fakestops=stops;
```

```
%rafinarea pe criteriile ratei de treceri prin zero
for i=1:len2
    a=starts(i);
    b=stops(i);
    %parcurgem in sens invers vectorul ratei de zero pana cand scade sub
    %valoarea de prag,cand creste peste prag acel punct devine startul
    %cuvantului
    c=0;stop=0;start=0;
    while start==0
        a=a-1;
        if ntz(a)>Ntzth
            c=c+1;
        else starts(i)=a;
            start=1;
        end
        %alegem pozitia in care pragul ntz este depasit de 15 ori
        if c==15
            starts(i)=a;
            start=1;
        end
        u=u+1;
        if u==CL starts(i)=a;start=1;
        end
    end
    %parcurgem in sens crescator vectorul ratei de zero pana cand scade sub
    %valoare de prag,cand creste peste prag acel punct devine capatul
    %cuvantului
    c=0;
    while (stop==0 & b < lenen)
        b=b+1;
        if ntz(b)>Ntzth%
           c=c+1;
        else stops(i)=b;
            stop=1;
        end
         %alegem pozitia in care pragul ntz este depasit de 2 ori
        if c==15
             stops(i)=b;
            stop=1;
        end
        u=u+1;
        if u==CL stops(i)=b;
            stop=1;
        end
        if b==len-1
            stop=1;
        end
    end
words{i}=inreg(starts(i)*dur:stops(i)*dur);
end
% inregistrarea formata doar din cuvintele obtinute in urma rafinarii
newinreg=words{1,1}';
for i=2:len2
newinreg=[newinreg words{1,i}'];
end
newinreg=newinreg';
% timpii de start si stop in secunde
```

```
timestarts=starts*dur/fs;
timestops=stops*dur/fs;
%inregistrarea formata doar din cuvintele obtinute inaintea rafinarii
for i=1:len2
    words2{i}=inreg(fakestarts(i)*dur:fakestops(i)*dur);
end
newinreg2=words2{1,1}';
%xax folosit pentru a afisa timpii de start si stop pe axe
xax=sprintf(' %d %d ',timestarts(1),timestops(1));
for i=2:len2
newinreg2=[newinreg2 words2{1,i}'];
xax=[xax sprintf(' %d %d ',timestarts(i),timestops(i))];
end
len4=length(newinreg2);
len3=length(newinreg);
% sound(inreg,fs)
% sound(newinreg,fs)

for i=1:len2
    d{i}=sprintf('%0.3f %s',timestarts(i),'s');
    e{i}=sprintf('%0.3f %s',timestops(i),'s');
end
timestarts=d;
timestops=e;

index_start=timestarts;
index_stop=timestops;
for i=1:length(index_start)

    a=inreg(starts(i)*dur:stops(i)*dur-1);
    str=sprintf('%s%s%s%s%s%d',dest,filename,'/',filename,'_',i);
    wavwrite(a,fs,str)
end
end

function [count]=octaveWavMerge(root,file,dest)
files=dir(root)
b=0;
for i=3:length(files)
  if strcmp(files(i).name,'.directory')==0
          a=wavread([root '/' files(i).name ]);
      b=[b;a]
  endif

endfor
str=sprintf('%s%d',file);
wavwrite(b,[dest '/' str]);
end
```

Anexa2

Script antrenare

```
enrollment.sh file folder
file="$1"
folder="$2"
file2=`echo $file | cut -c 1-9` #se taie doar primele 9 caractere după 0
var=$(grep $file2 test/trainModel.ndx) #se caută dacă este deja antrenat vorbitorul
if [ $(grep $file2 test/trainModel.ndx | wc -w) -gt 0 ];then
var2=$var" "$file  #in caz pozitiv se adaugă noua înregistrare la cele existente
echo $file >> test/files/lst/UBM.lst #se generează fișierele folosite în antrenare
echo $file >> test/files/lst/all.lst
echo $file >> test/files/ndx/totalvariability.ndx
echo "$file $file " >> test/files/ndx/ivExtractor.ndx
varPldaStored=$(grep $file2 test/files/ndx/Plda.ndx)
varPldaNew=$varPldaStored" "$file
varNormStored=$(grep $file2 test/files/ndx/ivNorm.ndx)
varNormNew=$varNormStored" "$file
sed -i "s/$varPldaStored/$varPldaNew/g" test/files/ndx/ivNorm.ndx
sed -i "s/$varNormStored/$varNormNew/g" test/files/ndx/Plda.ndx
sed -i "s/$var/$var2/g" test/trainModel.ndx
else echo $file2" "$file >> test/trainModel.ndx #dacă nu a fost deja antrenat
echo $file >> test/files/lst/UBM.lst #se adaugă noul vorbitor în coadă
echo $file >> test/files/lst/all.lst
echo $file >> test/files/ndx/totalvariability.ndx
echo "$file $file " >> test/files/ndx/ivExtractor.ndx
echo $file >> test/files/ndx/Plda.ndx
echo $file >> test/files/ndx/ivNorm.ndx
fi

destfolder=test/files/data/prm/
mfccfile=$file".mfcc"
echo "Extragere caracteristici: `date`"
CMD_EXTRACT_FEA="sfbcep -F PCM16 -l 10 -p 19 -e -D -A $folder/$file
$destfolder$mfccfile"
$CMD_EXTRACT_FEA
echo "Sfârșit extragere caracteristici: `date`"
echo
echo "Normalizare caracteristici: `date`"
CMD_NORM="test/bin/NormFeat --config test/cfg/NormFeat_SPro.cfg --
inputFeatureFilename $file --featureFilesPath test/files/data/prm/   --
labelFilesPath  test/files/data/lbl/"
echo $CMD_NORM
$CMD_NORM
echo "Sfârșit normalizare caracteristici : `date`"
echo
echo "Antrenare model universal folosind maximizarea EM `date`"
test/bin/TrainWorld --config test/cfg/TrainWorld.cfg  &>
test/files/log/TrainWorld.log
echo "            sfârșit, vezi log/TrainWorld.log pentru detalii `date`"
echo
echo "Calculul matricei de variabilitate `date`"
test/bin/TotalVariability --config test/cfg/TotalVariability_fast.cfg &>
test/files/log/TotalVariability.log
echo "            sfârșit, vezi log/TotalVariability.log pentru detalii `date`"
echo
echo "Extragere i-vectors `date`"
```

```
test/bin/IvExtractor --config test/cfg/ivExtractor_fast.cfg &>
test/files/log/IvExtractor.log
echo "            sfârșit, vezi log/IvExtractor.log pentru detalii `date`"
rm $folder/$file
```

Anexa3

Script testare

```
identification.sh file folder
file="$1"
folder="$2"
destfolder=test/files/data/prm/
echo "$file $file " >> test/ivExtractor.ndx #se generează fișierele pentru testare
echo "$file $file " >> test/ivNorm.ndx
echo $file >> test/files/lst/all.lst
mfccfile=$file".mfcc"

echo

echo "Extragere caracteristici: `date`"
CMD_EXTRACT_FEA="sfbcep -F PCM16 -l 10 -p 19 -e -D -A $folder/$file
$destfolder$mfccfile"
$CMD_EXTRACT_FEA
echo "Sfârșit extragere caracteristici: `date`"
echo
echo "Normalizare caracteristici: `date`"
CMD_NORM="test/bin/NormFeat --config test/cfg/NormFeat_SPro.cfg --
inputFeatureFilename $file --featureFilesPath test/files/data/prm/   --
labelFilesPath  test/files/data/lbl/"
$CMD_NORM
echo "Sfârșit normalizare caracteristici : `date`"

c=`cat test/trainModel.ndx | awk '{print $1}'` #se extrage numele modelelor
#se generează fișierul de testare

echo $file$" "$c >> test/files/ndx/ivTest_plda_target-seg.ndx

echo "Extragere i-vectors `date`"
test/bin/IvExtractor --config test/cfg/ivExtractor_fast.cfg &>
test/files/log/IvExtractor.log
echo "            sfârșit, vezi log/IvExtractor.log pentru detalii `date`"
rm $folder/$file

echo "Normalizare i-vectors `date` "
test/bin/IvNorm --config test/cfg/ivNorm.cfg &> test/files/log/IvNorm.log
echo "            sfârșit, vezi log/IvNorm.log pentru detalii `date`"

echo "Antrenează modelul PLDA `date`"
test/bin/PLDA --config test/cfg/Plda.cfg &> test/files/log/Plda.log
echo "            sfârșit, vezi log/Plda.log pentru detalii `date`"


echo "Comparare modele stocate folosind analiză PLDA `date`"
test/bin/IvTest --config test/cfg/ivTest_Plda.cfg --targetIdList
test/trainModel.ndx &> test/files/log/IvTest_Plda.log
```

```
echo "            sfârșit, vezi log/IvTest_Plda.log pentru detalii `date`"
truncate -s -"$(tail -n1 test/files/lst/all.lst | wc -c)" test/files/lst/all.lst
#extragere din fișierul de scoruri doar scorurile pentru a fi folosite în interfață
cat /home/Calin/Desktop/prototip/test/files/res/scores_PLDA_lengthnorm.txt | awk
'{print $5}' > /home/Calin/Desktop/prototip/test/files/res/scores

rm $folder/$file
```

Anexa 4

Calcul curbă ROC și matrice de confuzie

```
#exemplu baza TSP
load('/home/Calin/experimente/experimente_dizertatie/baza_TSP/files/res/scores')
k=0;p=0;pp=1;tp=zeros(1,351);fp=zeros(1,351);fn=zeros(1,351);tn=zeros(1,351);z=0;ma
t=zeros(23,23);
for i=1:length(scores)
k=k+1; #numără scorurile
if k==23 #la fiecare 23 de scoruri se incrementează o nouă înregistrare
p=p+1;
if p==28 #la fiecare 28 de înregistrări se incrementează un nou vorbitor
    pp=pp+1 #indicele vorbitorului
    p=0;
end
 for j=-20:0.1:10  #se balează pragul de la -20 la 10 cu un pas de 0.1
z=z+1;  #se reține poziția pragului
a=(find(scores(i-22:i)>j))'; #se caută scorurile mai mari decât prag pe fiecare
fereastră de 23 de scoruri
 if find(pp==a) #dacă a fost găsit indicele vorbitorului
 tp(z)=tp(z)+1; #se incrementează vectorul tp cu 1                    al pragului j
 fp(z)=fp(z)+length(a)-1; #se incrementează vectorul fp cu len(a) -1
 fp2=length(a)-1; #se salvează numărul de fp al pragului j
 fn(z)=fn(z)+0; #autentificările false nu se incrementează
 tn(z)=tn(z)+23-fp2-1; #tn se incrementează cu diferența între nr total de
posibilități și numărul de autentificări false și pozitive
 else fn(z)=fn(z)+1; #dacă nu a fost găsit indicele vorbitorului se incrementează
 fp(z)=fp(z)+length(a); vectorul de respingeri false cu 1
 fp2=length(a); , vectorul de autentificări false cu lungimea lui a
 tp(z)=tp(z)+0; vectorul de autentificări pozitive nu se incrementează
 tn(z)=tn(z)+23-fp2-1; vectorul de respingeri pozitive se incrementează cu
diferența dintre numărul total de posibilități și numărul de autentificări false și
pozitive
 end
end
 z=0; se reinitializează counterul pentru prag și pentru scoruri
 k=0;
end
end



frr=fn./(tp+fn) #se calculează far și frr pentru toate valorile pragurilor
far=fp./(tn+fp) #din graficul far/frr se extrage valoare pragului optim
```

```
k=0;p=0;tp=zeros(1,1);fp=zeros(1,1);fn=zeros(1,1);tn=zeros(1,1);z=0;mat=zeros(23,23
);cc=0;pp=1;
for i=1:length(scores)
k=k+1;
if k==23
p=p+1;
cc=cc+1;
if p==28
    pp=pp+1
    p=0;
end
j=-0.4 #pragul optim ales
z=z+1;
a=(find(scores(i-22:i)>j))';
c=scores(i-22:i)
for y=1:length(a)
mat(cc,a(y))=c(a(y));  # se completează matricea mat pe linia corespunzătoare
înregistrării testate cu valorile peste pragul optim
end
end
end
count=0;p=0;k=1;
for i=1:644
    for j=1:23
        p=p+1;
        if k==j && max(mat(i,:))==mat(i,j) #se verifică dacă pe diagonala
principală se află maximul liniei.
            count=count+1
        end
    end

    if p==644
        k=k+1;
        p=0;
    end
end
```