# AUTHENTICATION SYSTEM OF BANKING TRANSACTION BY FINGERPRINTING

## Neural Network Approach

**Eugène MBUYI MUKENDI[1], Jean Didier BATUBENGA MWAMBA[2]**

**[1] Professor University of Kinshasa, Computer Science Department, DR Congo**

**[2] University of Kinshasa, Computer Science Department, DR Congo**

**Kinshasa Computer Science Laboratory.**

**Abstract**

Shape recognition is a set of techniques and methods aiming at identifying patterns from raw data so as to make a decision depending on the category assigned to these patterns. Examples of content to which methods are applied are numerous. It can be visual content such as bar code, face, fingerprint, speech, images and much more. In fingerprint recognition, the application makes the registration of a person as well as his fingerprint in a database for future authentication, that is to say determine whether two fingerprints are identical to conclude that they come from the same person. In our case, the application authenticates the client of the Bank and then asks for his/her account number and password to ca rry out the withdrawal. The algorithm for fingerprint recognition which is used has been proposed by **D. Maio** and allows the location of minutiae in a more direct way by using neural networks. [2], [4]

Key words: authentication, fingerprinting, neural networks, identification.

## I. INTRODUCTION [1], [2], [3], [4]

### I.1. Pattern recognition

#### A. Kind of pattern recognition

In general, there are two types of pattern recognition:

- **Identification**: determines whether an observation or input data is a manifestation of the individual previously known.
- **Classification**: determines that the observation is a manifestation of a member of a class. In both cases, individuals or classes are characterized by a vector of properties. The recognition is summed up in a comparison of the properties.

#### B. Pattern recognition system

A pattern recognition system principally possesses an equipment of acquisition of the object to recognize. These equipments can be a video, a fingerprint reader, microphone etc. From them, the system acquires the input data that will undergo pretreatment to improve their qualities. Generally, this task is performed by hardware. After the preprocessing stage, the system goes to the extraction of features that will be used to identify the object to recognize. The next step will consist in the classification which aims, for a given input object, to specify which class it belongs to.

With the help of special equipment depending on the type of recognition to achieve, the acquisition of data to be processed can be manipulated by machine. The pattern to recognize is often a digitized image or a speech segment. It could undergo pretreatment so as to be ready for manipulation. The structure of a pattern recognition system contains the following items:

**Pretreatments**: have to determine hash information and retain the relevant one, useful for recognition. They can provide standardized data. They consist of noise reduction, the segmentation, normalization and skeletonization.

**Parameterization**: consists in the reduction of the complexity of data to be processed by some useful parameters stored in the database. The parameters will represent the actual data in an irrefutable way. They will be made thanks to characteristics extracted from data. In the parameterization, the system can calculate the type of representation, extract features and finally pass to learning.

**The post-processing**: is to validate the results of the classification using specific tools scope.

### I.2. Phases of pattern recognition

Applications of pattern recognition are made through a pre-registration of the form to be recognized. It is only from this record that the application may proceed to the recognition of the so called form by comparing it with those previously recorded.

### I.3. Algorithm for fingerprint recognition

Generally there are two categories of algorithms for fingerprint recognition: the first category concerns "conventional" algorithms that rely on the relative position between minutiae while the second includes algorithms to extract other features of the fingerprint, such as the local direction of the grooves, or

the local frequencial components of the texture in the heart of the image.
Note also the alternative proposed by **D. Maio** that allows the location of minutiae in a more direct way by using neural networks.

Figure 1. Fingerprints

A neural network is composed of a set of interconnected formal neurons giving rise to various network structures. For our application, we used a structure in successive layers (Multi-Layer Perceptron: MLP). Such a structure ( See Figure 3) broadcasts the information of the input layer, composed by the artificial neural receiving primitive information, to the output layer, which contains the final neurons transmitting output information processed by the all network while traversing one or more intermediate layers, said hidden layers. So established, the network is a non-linear system that combines the input feature vectors to the outline of the image of the output layer.

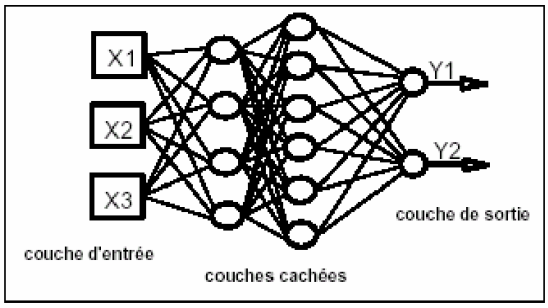


Figure 2. Multilayer neural network (also called Multi-Layer Perceptron MLP).

The neural network developed uses a threshold sigmoidal function and a gradient retropropagation to optimize its learning phase. Retro propagation consists in retropropagating error committed by a neuron's synapses and neurons that are connected. For neural networks, gradient error retropropagation are commonly used to correct errors in the importance of the elements that have just participated in the realization of these errors: the synaptic weights that contribute to generate an important error will be more significantly modified than weights which have led to a marginal error.

## II. MODELLING AND IMPLEMENTATION OF APPLICATION [4], [5] [6]

### II.1. Description of Application

The application that we are going to design will be responsible for authenticating a client prior to a transaction, especially in case of withdrawal. Being a biometric system, the application makes the customer's record as well as his/her fingerprint for future authentication. Thus customers register their fingerprints as well as useful information.

During withdrawal, the application authenticates the client then asks his/her account number and password to complete the withdrawal.

### II.2. Modeling the Application

#### II.2.1. Use case diagrams

##### II.2.1.1. Determination of system actors

Actors represent different entities that will interact with the system.

The different actors are:

- Fingerprint readers: the role is to capture customers' fingerprints
- Customer: Who can make a withdrawal or deposit
- The user creates the account of the customer, records the customer in the enrollment phase and receives deposits of money from different client. She/he administers the program

##### II.2.1.2. Description of use cases

In the table below we describe the different use cases used in our application.

##### II.2.1.3. Use case diagrams

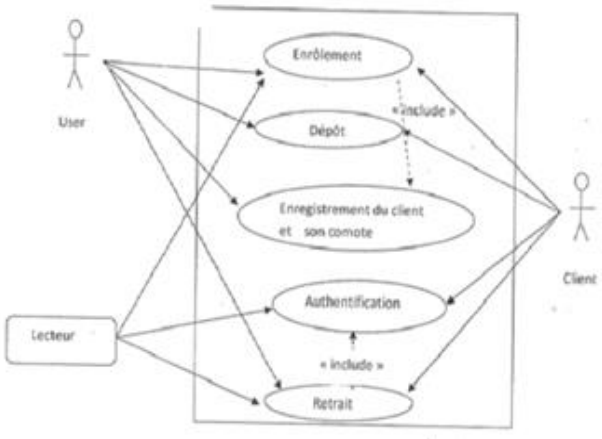


Figure 3. Use case diagrams

### II.3. Data base design

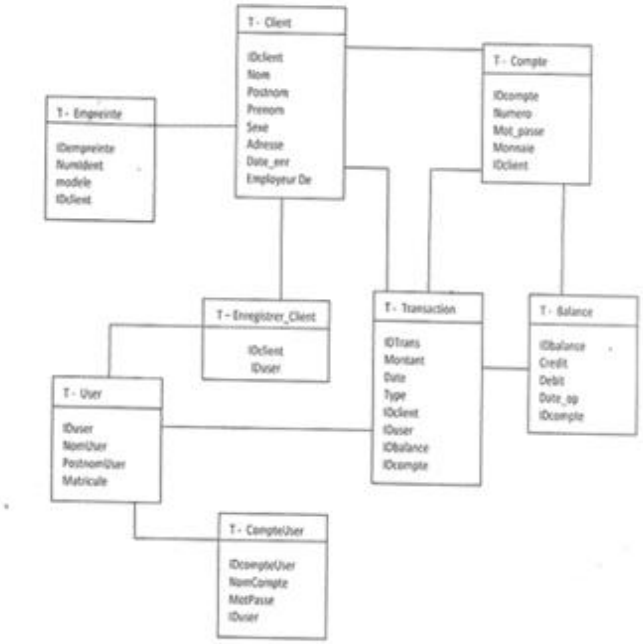


Figure 4. Passage from class diagram to MLDR

### II.4. Interface presentation for the Application







### II.5. Extract source code of the application

```
import java.awt.Dimension;
import java.awt.Toolkit;
import javax.swing.JOptionPane;
public class Appi extends javax.swing.JFrame {
  public Appi(String user) {
    initComponents();
     Dimension dim=Toolkit.getDefaultToolkit().getScreenSize();
    this.setLocation(dim.width/2-this.getWidth()/2,dim.height/2-this.getHeight()/2);
    txtCompte.setText(user);
  }
private void creerDossierClientActionPerformed(java.awt.event.ActionEvent evt) {
    Client monClient = new Client();
    panClient.setRightComponent(monClient);
  }
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    User user = new User();
    panConfig.setRightComponent(user);
  }
  private void bEnrollerActionPerformed(java.awt.event.ActionEvent evt) {
    FormMain classeEnroll= new FormMain();
    classeEnroll.setVisible(true);
  }
  private void bRetraitActionPerformed(java.awt.event.ActionEvent evt) {
 TransactionRetrait retrait = new TransactionRetrait();
    panTransact.setRightComponent(retrait);
     FormMain classeEnroll= new FormMain();
          classeEnroll.setVisible(true);

  }
  private void depotActionPerformed(java.awt.event.ActionEvent evt) {
    // depot
    TransactionDepot depot = new TransactionDepot();
    panTransact.setRightComponent(depot);
  }
  private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {
    Login login;
    int rep = JOptionPane.showOptionDialog(panClient,"Voulez vous réellement quitté", "Fermeture
du Programme",
JOptionPane.OK_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE,null, null,null);
    if(rep == JOptionPane.OK_OPTION)
      login = new Login();
  }
```

```
public static void main(String args[]) {
    final String compte = null;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Appi(compte).setVisible(true);
        }
    });
```

**Code pour le dépôt**

```
package contenu;
import java.io.FileInputStream;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.util.Date;
import java.util.HashMap;
import javax.swing.JOptionPane;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
public class TransactionDepot extends javax.swing.JPanel {
    Connecion connexion;
    double soldeAncien;
    int id,id_client,id_bal;
    public TransactionDepot() {
        initComponents();
        connexion = new Connecion();
    }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String nom = tnom.getText();
        String numero = txtNumeroCompte.getText();
        id = 0; double credit = 0,debit = 0;
        try{
            java.sql.Statement stmt=connexion.connect.createStatement();
            String requete="SELECT
'Nom','PostNom','Prenom','Adresse','Numero','Monnaie',IDcompte,client.IDclient " +
                "FROM 'client',compte " +
                "WHERE 'Numero'='"+numero+"' and client.Nom='"+nom+"'";
            ResultSet res=stmt.executeQuery(requete);
            while(res.next()){
                txtNom.setText(res.getString(1));
                txtPostnom.setText(res.getString(2));
                txtPrenom.setText(res.getString(3));
                tAdresse.setText(res.getString(4));
                jNumeroCompte.setText(numero);
                jMonnaie.setText(res.getString(6));
                id = res.getInt(7);
                id_client = res.getInt(8);
            }
            String reque="SELECT Credit,Debit,IDbalance FROM balance WHERE
balance.IDcompte='"+id+"'";
            ResultSet re=stmt.executeQuery(reque);
            while(re.next()){
                debit = re.getDouble(2);
                credit = re.getDouble(1);
                id_bal = re.getInt(3);
            }
            soldeAncien = credit - debit;
            jSoldeAncien.setText(""+soldeAncien);
        }catch(Exception ex){
            JOptionPane.showMessageDialog(null,"Erreur"+ex);
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // Crediter :le compte
        Date datejr = new Date();
        String date = DateFormat.getDateInstance().format(datejr);
        date =date+" a "+DateFormat.getTimeInstance().format(datejr) ;
        double soldeActuel;
        soldeActuel = soldeAncien + Double.parseDouble(txtMontant.getText());
        txtActuel.setText(""+soldeActuel);
        //processus de creditation du solde
        try{
        java.sql.Statement stm=connexion.connect.createStatement();
        String requete="insert into trasanction "
                +
"values(NULL,'"+Double.parseDouble(txtMontant.getText())+"','"+date+"','Depot','"+id+"'," +
                "'herve','"+id_client+"','"+id_bal+"')";
        int re=stm.executeUpdate(requete);
        String reque= "UPDATE balance SET Credit = Credit +
"+Double.parseDouble(txtMontant.getText())+" WHERE(IDcompte='"+id+"')";
        int r=stm.executeUpdate(reque);
        String reqet="SELECT MAX(IDtrans) FROM trasanction";
            ResultSet reponse =stm.executeQuery(reqet);
            reponse.next();
            JOptionPane.showMessageDialog(null,"Compte Crédité ! ");
            //impression du document
            String cheminFichier="C:\\Rapport\\InfoTransaction.jasper";
            FileInputStream cheminReport=null;
            cheminReport=new FileInputStream(cheminFichier);
```

```
            HashMap param=new HashMap();
            param.put("id_client",id_client);
            param.put("numero",jNumeroCompte.getText());
            param.put("id_trans",reponse.getInt(1));

            JasperPrint jprint=JasperFillManager.fillReport(cheminFichier, param,connexion.connect);
            JasperViewer.viewReport(jprint,false);
          }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Erreur"+ex);
      }
    }
    private void txtMontantFocusLost(java.awt.event.FocusEvent evt) {
      // calcul de la solde actuel
    }
```

**Code du le retrait**

```
package contenu;
import java.io.FileInputStream;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.util.Date;
import java.util.HashMap;
import javax.swing.JOptionPane;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
public class TransactionRetrait extends javax.swing.JPanel {
   Connecion connexion;
    double soldeAncien;
    int id,id_client,id_bal;
   public TransactionRetrait() {
      connexion = new Connecion();
      initComponents();
   }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      String motPasse= tnom.getText();
      String numero = txtNumeroCompte.getText();
      String numCompte="";
      id = 0; double credit = 0,debit = 0;

      try{
        java.sql.Statement stmt=connexion.connect.createStatement();
        String requete="SELECT
'Nom','PostNom','Prenom','Adresse','Numero','Monnaie',IDcompte,client.IDclient " +
            "FROM 'client',compte " +
            "WHERE 'Numero'='"+numero+"' and compte.MotPasse='"+motPasse+"' and
client.Idclient = compte.IDclient ";
        ResultSet res=stmt.executeQuery(requete);
        while(res.next()){
          txtNom.setText(res.getString(1));
          txtPostnom.setText(res.getString(2));
          txtPrenom.setText(res.getString(3));
          tAdresse.setText(res.getString(4));
          jNumeroCompte.setText(res.getString(5));
          jMonnaie.setText(res.getString(6));
          id = res.getInt(7);
          id_client = res.getInt(8);
          numCompte = res.getString(5);
        }
        String reque="SELECT Credit,Debit,IDbalance FROM balance WHERE
balance.IDcompte='"+id+"'";
        ResultSet re=stmt.executeQuery(reque);
        while(re.next()){
          debit = re.getDouble(2);
          credit = re.getDouble(1);
          id_bal = re.getInt(3);
        }
        soldeAncien = credit - debit;
        jSoldeAncien.setText(""+soldeAncien);

      }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Numero compte n'est pas valide : "+ex);
      }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // Crediter :le compte
      Date datejr = new Date();
      String date = DateFormat.getDateInstance().format(datejr);
      date =date+" a "+DateFormat.getTimeInstance().format(datejr) ;
      double soldeActuel;
      soldeActuel = soldeAncien - Double.parseDouble(txtMontant.getText());
      txtActuel.setText(""+soldeActuel);
      //processus de creditation du solde
      try{
        java.sql.Statement stm=connexion.connect.createStatement();
        String requete="insert into trasanction "
            +
"values(NULL,'"+Double.parseDouble(txtMontant.getText())+"','"+date+"','Retrait','"+id+"'," +
```

```
            "herve','"+id_client+"',"+id_bal+")";
        int re=stm.executeUpdate(requete);

        String reque= "UPDATE balance SET Debit = Debit +
"+Double.parseDouble(txtMontant.getText())+", Credit = Credit -
"+Double.parseDouble(txtMontant.getText())+" WHERE(IDcompte='"+id+"')";
        int r=stm.executeUpdate(reque);
        String reqet="SELECT MAX(IDtrans) FROM trasanction";
          ResultSet reponse =stm.executeQuery(reqet);
          reponse.next();
        JOptionPane.showMessageDialog(null,"Compte Débité ! ");

        //impression du document
         //impression du document
          String cheminFichier="C:\\Rapport\\InfoTransaction.jasper";
          FileInputStream cheminReport=null;
          cheminReport=new FileInputStream(cheminFichier);
          HashMap param=new HashMap();
          param.put("id_client",id_client);
          param.put("numero",jNumeroCompte.getText());
          param.put("id_trans",reponse.getInt(1));

        JasperPrint jprint=JasperFillManager.fillReport(cheminFichier, param,connexion.connect);
        JasperViewer.viewReport(jprint,false);
      }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Erreur"+ex);
      }
}
```

**Extrait code d'enrollement**

```
/
  public void enroll(int IDclient) {
    try {
      //Inserts the template on the database
      enrollStmt.setBinaryStream(1,new ByteArrayInputStream(template.getData()),
template.getData().length);
      // enrollStmt.setInt(2, template.getQuality());
      enrollStmt.setInt(2, IDclient);
      enrollStmt.executeUpdate();

      //Picks the ID generated for it.
      ResultSet rs = insertedIdStmt.executeQuery();
      rs.next();
```

## CONCLUSION

We presented a biometric authentication system that identifies customers by means of fingerprints before carrying out banking transactions while withdrawing money. A complete processing chain from the acquisition of fingerprints until the withdrawal has been developed with satisfaction. At the moment of enrollment, neural networks do learn the different fingerprints. During recognition, they combine a fingerprint stored in the database at the time of the learning process with the input in order to state the correspondence.

By adding authentication, notably by fingerprint during transactions we increase safety and minimize the risk of fraud and theft. The false acceptance rates and those of discharges of a potential client were minimized.

## REFERENCES

1. BELGUECHI R: Contribution to the fingerprint recognition by a hybrid approach INI, 2006.
2. CHAARI Anis: New approach to identification in biometric databases based on an unsupervised classification. PhD thesis, University of Enry, 2009.
3. DREYFUS G and others: Networks of neuron methodology and applications. Eyrolle 2004
4. MBUYI MUKENDI Eugene and others: Datamining and Neuronal networks I IJCSI, Volume 9, issue 5, september 2012.
5. MBUYI MUKENDI Eugène and others : Biometric system by pointing finger IJCSI volume 10, issue 4, juillet 2013.
6. MUNTENGE Kalili Herve: Automatic authentication system, UNIKIN 2010.