European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

BIOMETRIC AUTHENTICATION OF REMOTE FINGERPRINT LIVE SCAN USING ARTIFICIAL NEURAL NETWORK WITH BACK PROPAGATION ALGORITHM AND POSSIBILITY FOR WIDER SECURITY APPLICATIONS

*__Dr. Roseline Ifeoma Echezona,__
Library Department, University of Abuja, Nigeria, **__Dr. Oscar. N Atuma__.
Sociology department, University of Abuja

*** **Michael Umoh**.
ICT Unit, University of Abuja

******Prof. K.C. Ofokansi**,
Department of Pharmaceutics, University of Nigeria Nsukka

*******Orji Jolly Kalu**
Faculty of Physical Sciences, Department of Computer Science, University of Nigeria Nsukka.,

ABSTRACT: *This study is aim to experiments the development of an automated foolproof university library system that integrates fingerprint technique with fingerprint-based Personally Identified Number (PIN)/password architecture for enhanced registration and login security. The development environment for creating the electronic library application for universities as RESTful Web Service is Jersey Framework. This framework implements JAS-RX 2.0 API, which is a de facto specification for developing a RESTful Web Service-based software system. Other necessary programming technologies employed in the research work are JDK, Apache Tomcat and Eclipse, which were set up prior to setting up the Jersey Framework as the development environment. The study is therefore summarized by generating hash digital values of perfectly matched reference shape signatures formed from the extraction of global minutiae features, comparing and further matching each hash value with its corresponding highly encrypted password equivalence for unique establishment of a person's identity, minimal mean-*square *errors and unnecessary ambiguity introduced through false positives, as an extended security enhancement measure in biometric systems. , the study investigates the algorithm for generating templates for matching minutiae [10] together with the algorithm for generating reference axis [11], which infers that for a pair of minutiae (p_n , q_0) to match, there exists a reference point that corresponds between the two fingerprint images. The experimental result shows that the Sample fingerprint images were captured using a biometric scanner, which was integrated with the help of JAVA libraries, and stored in a database as raw image files..*

KEYWORDS: biometric, live scan, artificial neural network, rest architecture, algorithm

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

INTRODUCTION

Based on specificity, each conventional biometric system is designed for the purpose of unbiased recognition of a person's unique identity, perhaps, with the aid of complex mathematical models, computational algorithms, techniques and simulations, core fingerprint features which remain invariant, salient, distinct and natural to a person throughout his or her lifetime, such as colour of retina, pattern of voice, shape of face, shape of signature, image of fingerprint, etc. are unmistakably spotted out. A fingerprint image has many inherent features called minutiae. When the minutiae are extracted and separated into component features, it becomes safe to classify them into bifurcation, trifurcation, ridge ending, and sometimes, other global ridges such as shape signatures. Point pattern matching, graph matching and structural matching are popular methods for matching local minutiae features. The process of minutiae extraction, fingerprint verification and authentication is usually extensive and automated under artificial neural network training environment. Normally, the extracted features are classified into local and global features for matching purpose. The local features include minutiae points, which elicit immutable and distinct characteristics of the fingerprint image whereas the global features include basic ridge, pattern area, delta, type line, ridge count, etc. Some techniques for training fingerprint in an Artificial Neural Network (ANN) are Simplest Training Algorithm, the Newton's Algorithm, the Conjugate Gradient method, Quasi-Newton method, the Levenberg-Marquardt algorithm, Linear Vector Quantization (LVQ), one-dimensional optimization methods, multi-dimensional optimization methods, fuzzy logic, latent fingerprint, neuro fuzzy, etc. Artificial Neural Network, which can be described as computational model comprising a group of interconnected neurons for information processing, is adaptive and possesses the capability of transforming its arrangement at runtime in accordance with the input and output sequence that flows within the network. Artificial Neural Network can be trained from fingerprint samples fed into it in order to produce input/output matching, non-linearity, adaptivity, neuro biological analogy. The two most important types of biometric identification are concerned with behavioural features and physiological characteristics. The latter is concerned with palm-prints, finger-prints, retinal, iris, ear, face, handwriting, typewriting recognition, fraud detection, criminal sentencing, optical features and DNA verification.

Objective of the Study:

The primary objective of this study is to design, develop and implement a highly sophisticated electronic library system that authenticates the fingerprint of a library patron, and by extension any user, captured remotely by means of live scan with a biometric sensor attached to or built into a mobile or computing device.

- Another objective is to develop an automated foolproof university library system that integrates fingerprint technique with fingerprint-based Personally Identified Number (PIN)/password architecture for enhanced registration and login security.

Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Objectives of the Project were achieved with the following methodology:

The objectives of the research work were achieved by means of the stateless REpresentationational State Transfer (REST) architectural style, a RESTful Web Service, with Object-Oriented Analysis and Design (OOAD) methodology in conjunction its associated modelling tools from the Unified Modeling Language (UML) standard subset, namely class diagram, use case diagram, and sequence diagrams. The development environment for creating the electronic library application for universities as RESTful Web Service is Jersey Framework. This framework implements JAS-RX 2.0 API, which is a de facto specification for developing a RESTful Web Service-based software system. Other necessary programming technologies employed in the research work are JDK, Apache Tomcat and Eclipse, which were set up prior to setting up the Jersey Framework as the development environment. These were our conclusive results:

- The reasearch was approved by the Tertiary Education Trust Fund (TETFUND) as an Institution-Based Research grant awarded to our Research Group at the University of Abuja, Abuja under the scholarly leadership of the Chief University Librarian, Dr. Mrs. Ifeoma Roseline Echezona who adopted the work for implementation at Samuel Osaigbovo Ogbemudia main library using her good offices as her visionary library automation agenda. The highly sophisticated eLibrary software technology produced from this research work is being deployed at the library starting with the Office the University Librarian, Circulation Division, Aquisition Division, Technical Services Development Division, Special Collections Division, Serial Division, Readers Services Division to other Divisions sequencially. The automated system enables the university library users and administrators gain access to library facilities online through fingerprint cum username/password authentication, authorization and access routines.
- We are also integrating a new dimension of registraion into the system by means of fingerprint-based password authentication, authorization and access. This simply means that the username and password of a user are directly tied to his or her fingerprint image such that for every mismatch of fingerprint and password, the erring user will be thrown out by the system!
- The User Interface (UI) and User Experience (UX) of the new electronic library application offers exciting look and feel.

Theoretical Background

Another new approach to this research is the use of binarization to digitize minutiae values to form shaped signatures, which in turn, are matched with hash equivalence of fingerprint images stored in the relational knowledgebase. Re-Transitional State Transfer (REST) architecture is employed to handle HTTP requests from users and to establish the required communication exchange between the user and the remote knowledgebase using REST global variables. Our programming paradigm is an infusion Model View Controller (MVC) and JAVA Application Programming Interface (API), JAVA libraries and S Development Kit (SDK) and offers connectivity to the fingerprint scanner sensor.

Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

There is a synergy of PHP, JAVA applet, Tomcat server, and fingerprint knowledgebase using modularity programming over REST architecture.

Introduction of New Development Concepts With RESTful Architectural Style Using Java Application Progrogamming Interface (API), JDK, Tomcat and Eclipse

The architectural design of the electronic library system is shown fig. 2. The system was developed with two major server-side programming languages namely Hypertext Pre-Processor (PHP5) and Java. PHP is integrated with Java through various Java-based Application Programming Interfaces (APIs). PHP handles core operations of the system defined in well-structured classes, which spans across object-oriented needs of the clients (patrons, stakeholders, library, govenment agencies, etc.) as described in the GUI of the system (see above). PHP also communicates with MYSQL database server through the XAMP Server as localhost whereas Java communicates with the same MYSQL database server through the Apache Tomcat as it own localhost. The Java Development Kit (JDK) employed in the development of the elibrary system is jdk1.7.0_75. For strict purpose of software production, the Eclipse is used as the Integrated Development Environment (IDE). The version of JAX-RS is jaxrs-ri-2.17. , which is bundled with its deafault API directory, LIB directory (contains all the necessary Jersey libraries: C:\jaxrs-ri-2.17\jaxrs-ri\lib as well as another directory containing Jersey dependecies (C:\jaxrs-ri-2.17\jaxrs-ri\ext.

New Constraints To Contend With Under REST Architectural Style:

The research is contending with the six constraints of REST, namely: Uniform Interface, Stateless, Cacheable, Client-Server, Layered Sytem and Code On Demand.

Examination of Different Training Techniques For An Artificial Neural Network And Justification for Choosing Error Back Propagation Algorithm:

Artificial Neural Network is a mathematical model for simulating the activities of human brain based on its structure. The neural network processes fingerprint data or feature, called minutiae, fed into it one at a time and learns the features by classifying and comparing them. Usually, some errors are recorded during the initial classification phase which are fed or propagated back into the neural network and used for updating the algorithm of the network. This process or iteration goes on and on untill a desired output value is achieved from the output layer of the neural network. Hence, the term, error back propagation and the algorithm that propagates these errors is refered to as the Error Back Propagation Algorithm. An Artificial Neural Network is a multi-layered mapping network comprising interconnection of input, hidden and output layers modeled to intelligently train fingerprint image extracts using different supervised methods, techniques or algorithms such as the following training algorithms as summarized by [4]:

1. **Simplest Training Algorithm** – This first order training algorithm uses the *gradient descent or steepest descent* that requires information from a gradient vector. If we represent an artificial neutron as a function of weights, then $f(w_i) = f_i$ and $\nabla f(w_i) = g_i$. Starting from a point w_0 , this algorithm proceeds from w_i to a new point w_{i+1} and will not

Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

stop until a stopping criterion is satisfied in the training direction $d_i = -g_i$, which thus iterates, and this the training algorithm adopted for our experiment because it functions closely with the error back propagation algorithm.

$w_{i+1} = w_i - g_i \cdot \eta_i$, $i=0,1,\ldots$, where η is the training rate.

2. The Newton's method – This is a second order algorithm that uses the Hessian matrix and aims at finding better training directions by using the second derivatives of the loss function. Meanwhile the first and second derivatives of the loss function are denoted by:

$$\nabla_i f(w) = df/dw_i \quad (i = 1,\ldots,n)$$

and

$$H_{i,j} f(w) = d^2 f/dw_i \cdot dw_j \quad (i,j = 1,\ldots,n)$$

Representing $f(w_i) = f_i$, $\nabla f(w_i) = g_i$ and $Hf(w_i) = H_i$, we consider the quadratic approximation of f at w_0 using the Taylor's series expansion

$$f = f_0 + g_0 \cdot (w - w_0) + 0.5 \cdot (w - w_0)^2 \cdot H_0$$

H_0 is the Hessian matrix of f evaluated at the point w_0. By setting g equal to 0 for the minimum of f(w), we obtain the next equation

$$g = g_0 + H_0 \cdot (w - w_0) = 0$$

Therefore, starting from a parameter vector w_0, Newton's method iterates as follows

$$w_{i+1} = w_i - H_i^{-1} \cdot g_i, \quad i=0,1,\ldots$$

The vector $H_i^{-1} \cdot g_i$ is known as the Newton's step. Note that this change for the parameters may move towards a maximum rather than a minimum. This occurs if the Hessian matrix is not positive definite. Thus, the function evaluation is not guaranteed to be reduced at each iteration. In order to prevent such troubles, the Newton's method equation is usually modified as:

$$w_{i+1} = w_i - (H_i^{-1} \cdot g_i) \cdot \eta_i, \quad i=0,1,\ldots$$

The training rate, η, can either be set to a fixed value or found by line minimization. The vector $d = H_i^{-1} \cdot g_i$ is now called the Newton's training

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

direction. The study cannot adopt this training method because it requires a certain training direction and may tend towards maximum point with bias.

3. The conjugate gradient method: This algorithm functions as part of the gradient descent algorithm and the Newton's algorithm is motivated by the desire to accelerate the typically slow convergence associated with gradient descent. This method also avoids the information requirements associated with the evaluation, storage, and inversion of the Hessian matrix, as required by the Newton's method. In the conjugate gradient training algorithm, the search is performed along conjugate directions, which produces generally faster convergence than gradient descent directions. These training directions are conjugated with respect to the Hessian matrix. Representing d as the training direction vector, then, starting with an initial parameter vector w0 and an initial training direction vector d0 = -g0, the conjugate gradient method constructs a sequence of training directions as:

$$d_{i+1} = g_{i+1} + d_i \cdot \gamma_i, \quad i=0,1,\ldots$$

Here γ is called the conjugate parameter, and there are different ways to calculate it. Two of the most used are due to Fletcher and Reeves and to Polak and Ribiere. For all conjugate gradient algorithms, the training direction is periodically reset to the negative of the gradient. The parameters are then improved according to the next expression. The training rate, η, is usually found by line minimization.

$$w_{i+1} = w_i + d_i \cdot \eta_i, \quad i=0,1,\ldots$$

The study does not consider this training method most suitable even though it has the properties of the *gradient descent or steepest descent* algorithm.

4. Quasi-Newton method- This training algorithm (also generally referred to as the variable matrix methods) solves the common problem associated with the expensive computational cost of the Newton's algorithm, which requires many operations in order to evaluate the Hessian matrix and compute its inverse. Quasi-Newton approximates the inverse Hessian at each iteration of the algorithm using only information on the first derivatives of the loss function rather than calculate the Hessian directly and then evaluating its inverse. The Hessian matrix is composed of the second derivatives of the loss function [4]. The main idea behind the Quasi-Newton method is to approximate the inverse Hessian by another matrix G, using only the first partial derivatives of the loss function. Then, the quasi-Newton formula can be expressed as:

$$w_{i+1} = w_i - (G_i \cdot g_i) \cdot \eta_i, \quad i=0,1,\ldots$$

The training rate η can either be set to a fixed value or found by line minimization. The inverse Hessian approximation G has different flavours. Two of the most used are the Davidon–Fletcher–Powell formula (DFP) and the Broyden–Fletcher–Goldfarb–Shanno formula (BFGS). This is the default method to use in most cases: It is faster than gradient descent and conjugate gradient, and the exact Hessian does not need to be computed and inverted. This study would not choose this training method because it requires another matrix for final computation of the first matrix.

5. Levenberg-Marquardt algorithm

The Levenberg-Marquardt algorithm, also known as the damped least-squares method, has been designed to work specifically with loss functions which take the form of a sum of squared errors. It works without computing the exact Hessian matrix. Instead, it works with the gradient vector and the Jacobian matrix. Consider a loss function which can be expressed as a sum of squared errors of the form

$$f = \sum e_i^2, \quad i=0,\ldots,m$$

Here m is the number of instances in the data set. We can define the Jacobian matrix of the loss function as that containing the derivatives of the errors with respect to the parameters,

$$J_{i,j} f(w) = de_i / dw_j \quad (i = 1,\ldots,m \ \& \ j = 1,\ldots,n)$$

Where m is the number of instances in the data set and n is the number of parameters in the neural network. Note that the size of the Jacobian matrix is m·n. The gradient vector of the loss function can be computed as:

$$\nabla f = 2 J^T \cdot e$$

Here e is the vector of all error terms. Finally, we can approximate the Hessian matrix with the following expression.

$$Hf \approx 2 J^T \cdot J + \lambda I$$

Where λ is a damping factor that ensures the positiveness of the Hessian and I is the identity matrix. The next expression defines the parameters improvement process with the Levenberg-Marquardt algorithm

$$w_{i+1} = w_i - (J_i^T \cdot J_i + \lambda_i I)^{-1} \cdot (2 J_i^T \cdot e_i), \quad i=0,1,\ldots$$

When the damping parameter λ is zero, this is just Newton's method, using the approximate Hessian matrix. On the other hand, when λ is large, this

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

becomes gradient descent with a small training rate. The parameter λ is initialized to be large so that first updates are small steps in the gradient descent direction. If any iteration happens to result in a failure, then λ is increased by some factor. Otherwise, as the loss decreases, λ is decreased, so that the Levenberg-Marquardt algorithm approaches the Newton method. This process typically accelerates the convergence to the minimum.

A typical state diagram for the training process of a neural network with the Levenberg-Marquardt algorithm would show that the first step is to calculate the loss, the gradient and the Hessian approximation. Then the damping parameter is adjusted so as to reduce the loss at each iteration.

As we have seen the Levenberg-Marquardt algorithm is a method tailored for functions of the type sum-of-squared-error. That makes it to be very fast when training neural networks measured on that kind of errors. However, this algorithm has some drawbacks. The first one is that it cannot be applied to functions such as the root mean squared error or the cross entropy error and this the reason this study would not adopt it. Also, it is not compatible with regularization terms. Finally, for very big data sets and neural networks, the Jacobian matrix becomes huge, and therefore it requires a lot of memory. Therefore, the Levenberg-Marquardt algorithm is not recommended when we have big data sets and/or neural networks. The mathematical analysis of the above five training algorithms is an expression of [4].

Sigmoid (Activation) Function of the Artificial Neural Network

In human brain, the typical excitatory and inhibitory postsynaptic potentials at neural dendrites are represented by an artificial neuron in an artificial neural network, which sums up the weighted inputs it receives through a non-linear function referred to as the *activation* or *transfer* function, and in this study, a *sigmoid function*, which is a threshold function necessary for the activating of the training prcesses of minutiae features. For further explanations, a **sigmoid function** is a mathematical function and falls in the domain of real numbers with return value usually monotonically increasing from 0 to 1 or alternatively from −1 to 1, depending on convention and it is characterized with an S-shaped curve or **sigmoid curve whose special application is also found in** Gompertz curve (used in modeling systems that saturate at large values of x) and the ogee curve (used in the spillway of some dams), in statistics, the logistic distribution, normal distribution, *t* probability density functions, hyperbolic tangent, cumulative distribution functions, and in fact, other wider mathematical and computational applications A neuron in an artificial neural network is a set of input values (xi) and associated weights (wi) and a function (g) that sums the weights and maps the results to an output (y).

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

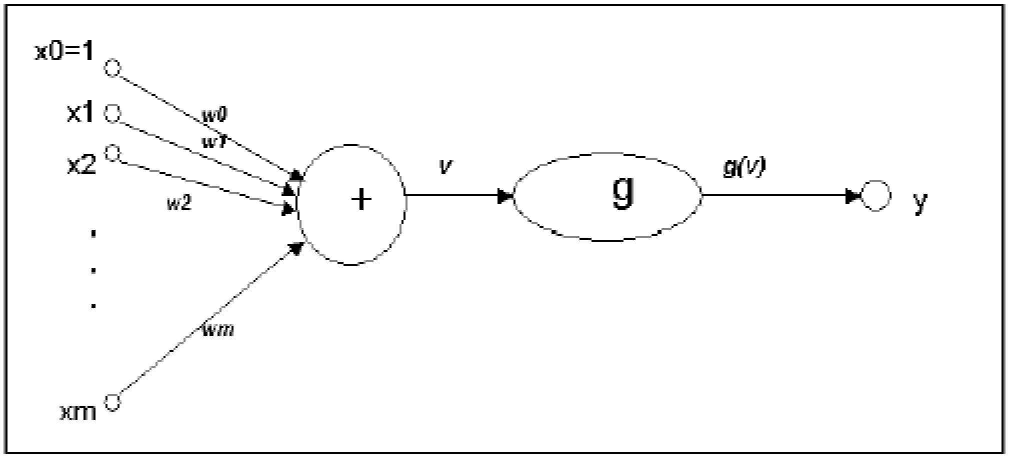


Fig. 1 Neurons are organized into layers (curled from [4]).

Fingerprint Knowledgebase Schema and Storage of Minutiae:
Let *e* denote a fingerprint extraction of a person *p* from a remote biometric live scan stored in a relational knowledgebase *k* of a multi-dimensional 16 x 16 pattern array whose data structure ranging from [k: 0, 1, 2, …, 255] accepts digital values of fingerprint minutiae including bifurcation, trifurcation, ridge ending and other global ridge extractions. Primarily, these local minutiae, noise and a few global ridges can be represented by:

$p[e][k]$ --- (1) , where the training for the first person is given by:

$p_1[e_1][k_1]$, k_1 denotes the field in the knowledgebase where fingerprint extraction, e_1, for the first person, p_1, is stored; and the training for the second person is expressed thus:
$p_2[e_2][k_2]$, k_2 denotes the field in the knowledgebase where e_2 is stored; and the general expression can be safely derived thus:

$p_n[e_n][k_n]$ --- (2)., where e_n is the 256th fingerprint extraction corresponding to the maximum storage capacity of the data structure defined in the by 16 x 16 pattern array. Similarly, let *d* denote the distance between a remote fingerprint sensor *s* and a centralized optical knowledgebase *k* running on JAVA apache tomcat such that for each iteration of fingerprint extraction from *p*, we have:

$p[e_i] \vee i \leq e \leq k$, where $\sum_{k=1}^{256} e_i$

It is known in the fingerprint industry that an artificial neural network is a mapping of binary values of fingerprint features fed into the neural network for training through its input layer whereas the corresponding weighted values $w(e_i)$ and threshold values for the PEs in its hidden layer are initialized with random numbers and then constantly adjusted

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

in each iteration till the desired target value is achieved in the output layer of the network. The adjusted weights, $w(e_i)$, form a new set of input values for further fingerprint training for verification or matching purpose. Errors usually occur after each training iteration, which is propagated back into the neural network. The study applies a sigmoid function for activation of pattern number and error back propagation algorithm as a training method in order to justify the accuracy of the ownership of a certain fingerprint of *p*. Using Levenborg Marquart Back Propagation Algorithm with no computation of Hussein Matrix as a second order training technique, [G] estimates Hussein matrix if performance functions having a form of sum of squares, thus:

H = JTY with calculation of gradient as:

g = JT, where "e" denotes vector of network error, J denotes Jacobian matrix that includes first order derivatives of network errors with respect to biases and words. G uses the standard BPA method for estimation of Jacobian matrix which is less difficult than the Hussein matrix estimation. But first, we have to establish the input patterns from the input vector, *p*[*e*][*k*], thus:

$Sum_{ej} = \sum w_{kj}\ U_{ei,}$ where U_{ei} represents the unit input for patter number *e*.
Adding up a weighted value w_{kj} to its corresponding threshold *t* in the hidden layer:
$\sum U_{ei} + t_j$, where t_j represents the sum of weight and threshold for the jth PE in the hidden layer ……………………………………………………………………………………(3)

We need to determine the binary value of each PE output, i.e. whether it is 1 or 0:
$U_{ei} = 1/(1 + \exp^{-k \times Sum_{ej}})$, where k represents the spread factors and now U_{ei} becomes our new input for the next training ………………………………………………….. (4)
To find the computational error for feed forward operation, with compare the value of the output with the value of the target match: $A_{ak} = t_{ek} - U_{ek}$
Adjusting the weight vector with new values:
$\Delta w_{kj} = \eta_2 k_2 \delta_{ak} U_{ej}(1 - U_{ek})$, …………………………..(5)
where the derivative of the sigmoid function is represented by the integral
$\int(A_{ek}) = k/(1 – U_{ej})$
The desired fingerprint match is achieved in the output layer through the combinatory computation of the weight vectors W_{kj} :
$\Delta t_{ek} = \eta_e k_e (1 - U_{ek}) Ϗ \delta_{ek} U_{jk}$, …………………………..(6)

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

3-Layered Artificial Neural Network:

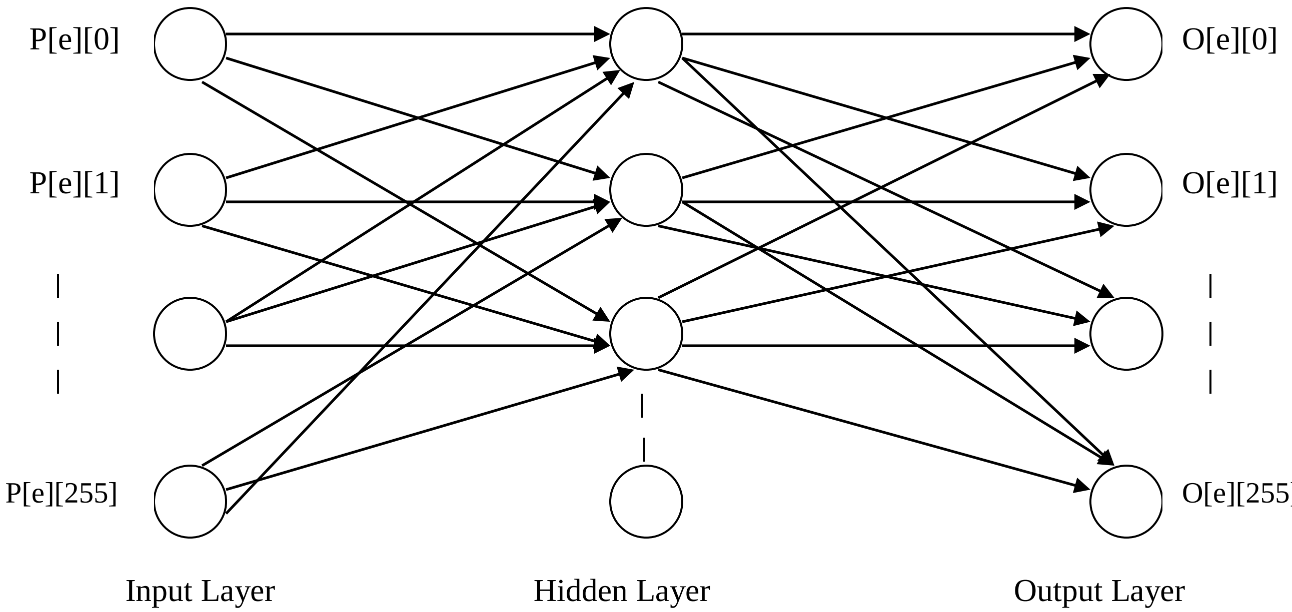


Fig.2 3-Layered Artificial Neural Network

Figure 1.0 represents the architecture of the proposed biometric-driven electronic library system. The fingerprint images of library patrons are captured with a biometric scanner and the extracted minutiae stored directly in the local computer to which the scanner is attached.

Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

THE RESTFUL DESIGN ARCHITECTURE

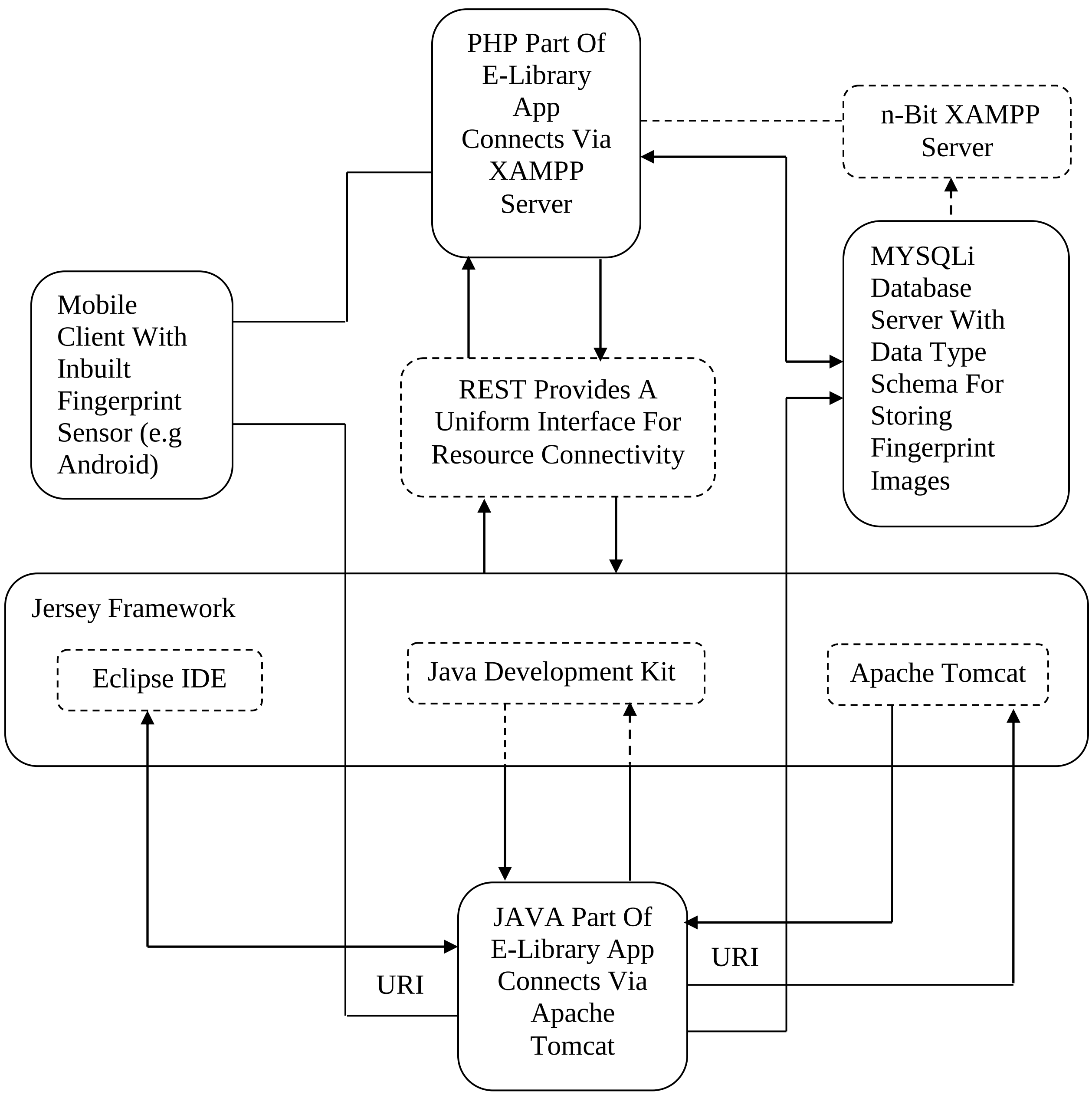


Fig. 4. Resource-Oriented Architecture (ROA) of a RESTful e-Library Web Service

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

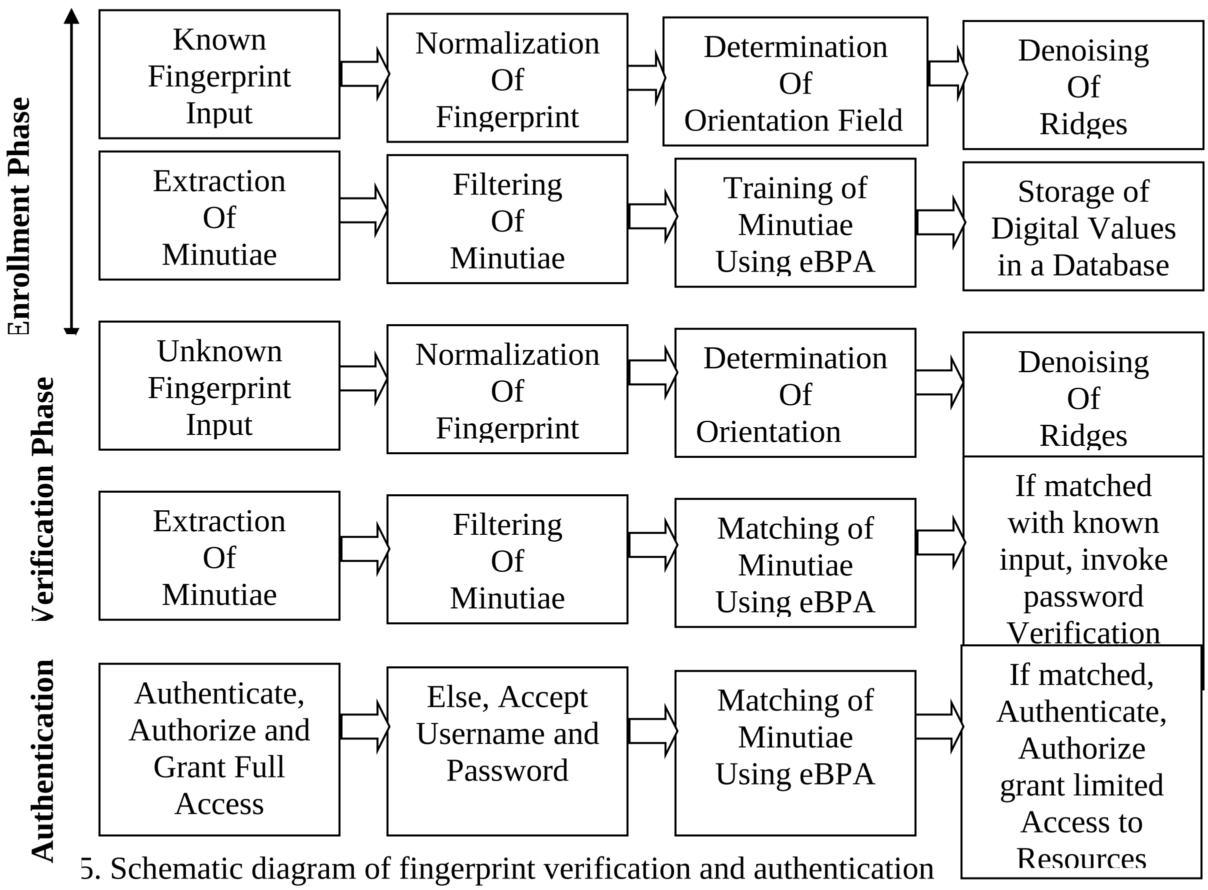
The Software Architecture

The software architecture of this system is made up of the view logic layer, controller logic layer and model logic layer, respectively. The Model View Controller programming concept was used in the development of the system. The brief explanation will prove useful:

The Model View Controller Logic

The system uses biometric sensing device to capture fingerprint images of a university library patron using Model – View – Controller programming design with the concept of separation of concerns. The system consists of three major and well-defined structures that are categorized into the view logic, the controller logic and the model logic respectively. The view logic layer of this software system is like the typical presentation layer of most software applications except additional work is done in the design and coding to separate concerns and protect the system against hackers and external site scripting attempts. The view logic represents the graphical user interface of the system, which provides an attractive but simple user-friendly interface for capturing fingerprint of a user through a biomtric sensing device as a prerequisite for logging into a university library system and membership registration. All the sofwtare components that are responsible for processing requests made by library users are separated from the view logic layer and dumped in the controller logic layer. The controller logic is an intermediary interface that lies between the view logic and the model logic. It coordinates all the operations that take place between the view logic and the model logic in a very intelligent sequence. It renders a user request such as captured fingerprint image made in the graphical user interface (i.e. the view logic) in form of interaction to appropriate sub-structures of the model logic. The study further divides the controller logic into the front controller and the action controller. The front controller sub-structure of the controller logic is directly responsible for interacting with the user via the view logic whereas the action controller directly updates the model logic with user requests. When a request is made by a reader who either simply enters a keyword in the search field or clicks on a bibliographic index, the model logic, which directly interacts with the database engine,

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.1-15, February 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)



5. Schematic diagram of fingerprint verification and authentication

Experimental Result:
Sample fingerprint images were captured using a biometric scanner, which was integrated with the help of JAVA libraries, and stored in a database as raw image files. The study used Jersey framework, which integrates with Eclipse IDE, Java Development Kit and Apache Tomcat. The JAVA SDK connects the electronic library application to the REST architecture through URL. PHP serves user HTTP requests to XAMPP server during development phase and then to online fingerprint database. RE-transitional State Transfer (REST) provides the user interface between the system the containing and the central knowledgebase. A user, irrespective of his or her geographical location, downloads the application developed out of this research, thumb prints through a fingerprint interface of the application and gains login authentication into the remote automated library system. A bibliophile will have to register with the eLibrary software system at first instance, do series of check-ins from borrowing and returning books, confirming membership, defaulters and other eligibility status to generating computerized identity cards from Okportal Superstructure engine, and engaging in final year library clearance, not forgetting routine activities carried out electronically in various divisions of the library, at each processing point, the biometric module would checkmate any potential gait activity

which may be demonstrated in user intrusion behavior through typing, submitting scanned images other than fingerprints, and in future implementation, voice/speech pattern recognition and gesture will demand serious attention
.

References

1 .Ahmed, F & Amiruyyaman, M.(2003) "Fingerprint Authentication System Using Back-Progation", Accessed fromhttps://www.researchgate.net/publication/228433611_Fingerprint_Authentication_System_Using_Back-Propagation/figures?lo=1 on 2/10/2019
2. Jayaraman, B. Puttamadappa, C. Anbalagan, E. ,Mohan, E & Madane, S.(2008)Fingerprint Authentication Using Bac-Progation Algorithm . International Journal of Soft Computing 3(4) 282-287.
3 .Mamuno, M .& . Akatar, A.K.M. (2006). Fingerprint Verification System Using Artificial Neural Network. Information Technology Journal 5(6): 1063-1067,
4. https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network