Vol 7. No. 3 - September, 2014
African Journal of Computing & ICT

www.ajocict.net

Forged Signature Detection Using Artificial Neural Network

T. O. Oladele, K. S. Adewole & T. N. Abiodun
[1,2,4] Department of Computer Science
University of Ilorin
Ilorin, Nigeria.
tinuoladele@gmail.com, adewole.ks@gmail.com, abitess@live.com
+234 8164916082 +234 8038544348 +234 8054571566

A. O. Oyelami
Department of General Studies
Nigerian Institute of Jouralism
Ogba Estate, Ikeja, Lagos.
08walexy@gmail.com
+234 8038341776

ABSTRACT

Crimes and corruptions are practices that gradually cripple the economy of a nation most especially in Nigeria. Nigerian government has strived hard to reduce these acts perpetrated by the citizens. This is evident in the struggles of Economic and Financial Crime Commission (EFCC) and Independence Corrupt Practices and other Related Offences Commission (ICPC) to reduce frauds in both public and private sectors due to signature forgery which attempts to commit financial crimes and other related offences. Forged signature is an illegal copy of signature that looks like a genuine signature usually used for financial fraud. Identity verification (authentication) in computer systems has been traditionally based on something that one has such as key, magnetic or chip card or that one knows such as PIN or password. Things like keys or cards, however, tend to get stolen or lost and passwords are often forgotten or disclosed. In this paper, a neural network algorithm was employed to develop a system that can verify and detect forged signatures. The effect of the signature verification and detection is discussed and its impact on the economy is highlighted. Result of the proposed Java application shows its capability in detecting forged signatures. The system has the capability to learn from previous data and to assist EFCC and ICPC in detecting and investigating fraudulent activities.

Keywords: Neural Network, Algorithm, Biometrics, Signature & Forgery..

African Journal of Computing & ICT Reference Format:
T. O. Oladele, K. S. Adewole, T. N. Abiodun & A. O. Oyelami (2014). Forged Signature Detection Using Artificial Neural Network. Afr J. of Comp & ICTs. Vol 7, No. 3. Pp11-20.

1. INTRODUCTION

The purpose of this paper is to provide a fast, safe and easy to use application to detect forged signatures and this can be achieved by using several techniques but in this paper, neural network was employed being one of the techniques of Data mining. Data mining is the process of analyzing and extracting knowledge from large volume of data. Data mining can also be called Knowledge Discovery from Data (KDD). There is always need for signature verification to be carried out before any payment is made, say in banks, but this is done manually by making comparison between the signature presented and the one in the database in which only eyes cannot detect the authenticity of the signature. The signature of a person is an important biometric attribute of a human being which can be used to authenticate human identity [1, 2]. As signatures continue to play a very important role in financial, commercial and legal transactions, truly secured authentication becomes more and more crucial.

Handwritten signatures are considered as the most natural method of authenticating a person's identity. A signature by an authorized person is considered to be the "seal of approval" and remains the most preferred means of authentication. However, human signatures can be handled as an image and recognized using computer vision and neural network as a technique of data mining. With modern computers, there is need to develop fast algorithms for signature recognition [2]. This application will analyze, compare, mine, and evaluate patterns then present knowledge of the mined data. In a broad sense data mining is the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories.

Signature is a special case of handwriting which includes special characters and flourishes. Many signatures can be unreadable. They are a kind of artistic handwriting objects. However, a signature can be handled as an image, and hence, it can be recognized using computer vision and artificial neural network techniques. Signature recognition and verification involves two separate but strongly related tasks: one of them is identification of the signature owner, and the other is the decision about whether the signature is genuine or forged [3]. Considering the importance of Data mining, it is pertinent to study the manual method of signature verification. This paper dealt with the problem of verifying the authenticity of signatures manually and avoid the problems which occur when carried out manually. Some of the drawbacks of the existing system were identified and a computerized system that will be compatible with the existing system and be more user-friendly was designed. In this paper, the back propagation training algorithm of neural network was employed in the development of a system that can verify and detect forged signatures. The Forged Signature Detection System was implemented by using Java programming language.

2. RELATED WORKS

A number of biometric methods have been introduced, but few have gained wide acceptance. This approach, concentrates on some techniques to gather knowledge. Adrian Perrig introduces the BiBa signature scheme. BiBa stands for Bins and Balls signature. A collision of balls under a hash function in bins forms the signature [4]. BiBa signature scheme is a new signature construction that uses one-way functions without trap doors. The most important features of BiBa signature scheme is a low verification overhead and a relatively small signature size. In comparison to other one-way function based signature schemes, BiBa has smaller signatures and is at least twice as fast to verify (which probably makes it one of the fastest signature scheme to date for verification). Ihler et al presented modeling dynamical processes. They used this methodology to model handwriting stroke data, specifically signatures, as a dynamical system and show that it is possible to learn a model capturing their dynamics for use either in synthesizing realistic signatures and in discriminating between signatures and forgeries even though no forgeries have been used in constructing the model [5]. Another study reported in [6] explained the method for on-line signature authentication, which is based on an event-string modeling of features derived from pen-position and pressure signals of digitizer tablets. The intra- class variability is one of the key problems in biometrics.

Pivonka, and Nepevny, proposed a new controller by combination of Generalized Predictive Control (GPC) algorithm and Neural Network model, with many advantages. Neural model has the ability to observe system changes and adapted itself, therefore regulator based on this model also will be adaptive. Algorithm was implemented in MATLAB-Simulink with aspect of future implementation to Programmable Logic Controller (PLC).

It was tested on mathematical and physical models in soft-real-time realization. Predictive controller in comparison with classical PSD controller and the advantages and disadvantages were shown [7]. Plamondon and Lorette made an assertion that most of the work in off-line forgery detection has been on random or simple forgeries and less on skilled or simulated forgeries [8]. Papadimitrioy and Terzidis, proposed a fuzzy system that approximated the accurate set of rules keeping only the more important aspects of the data. The approximation algorithms either received an a priori description of a set of fuzzy sets or, especially for the case when interpretable fuzzy sets could not be pre-specified by the experts, the algorithm presented for building them automatically. After the construction of the interpretable fuzzy partitions, the developed algorithms extract from the SVFI rules a small and concise set of interpretable rules.

Finally, the Pseudo Outer Product (POP) fuzzy rule selection ordered the interpretable rules by using a Hebbian like evaluation in order to present the designer with the most capable rules [9]. Unluturk et al, conducted an extensive study on the use of Neural Network in Biometric researches. They developed the emotion recognition neural network (ERNN) to classify the voice signals for emotion recognition [10]. Dule, et al, worked on outdoor vehicle images to determine the color of the vehicle and color classes. In this work, the Performances of different feature sets obtained by various color spaces and different classification methods were taken into account in order to improve the outdoor vehicle color recognition [11].

2.1 Types Of Forgeries

The main task of any signature verification system is to detect whether the signature is genuine or counterfeit. Forgery is a crime that aims at deceiving people. Since actual forgeries are difficult to obtain, the instrument and the results of the verification depend on the type of the forgery [1, 12]. Basically there are three types that have been defined:

1. **Random forgery**: this can normally be represented by a signature sample that belongs to a different writer i.e. the forger has no information whatsoever about the signature style and the name of the person.
2. **Simple forgery**: this is a signature with the same shape or the genuine writer's name.
3. **Skilled forgery**: this is signed by a person who has had access to a genuine signature for practice [1, 8, 13].

2.2 Overview of Neural Networks

Neural networks offer a mathematical model that attempts to mimic the human brain. Knowledge is represented as a layered set of interconnected processors, which are called neurons. Each node has a weighted connection to other nodes in adjacent layers. Individual nodes take the input received from connected nodes and use the weights together with a simple function to compute output values. Learning in neural networks is accomplished by network connection weight changes while a set of input instances is repeatedly passed through the network.

Vol 7. No. 3 - September, 2014
African Journal of Computing & ICT

www.ajocict.net

Once trained, an unknown instance passing through the network is classified according to the values seen at the output layer. Surveys existing work on neural network construction, attempting to identify the important issues involved, directions the work has taken and the current state of the art. Neurons only fire when input is bigger than some threshold. It should, however, be noted that firing doesn't get bigger as the stimulus increases, it is an all or nothing arrangement. A typical neuron collects signals from others through a host of fine structures called dendrites.

The neuron sends out spikes of electrical activity through a long, thin stand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes [1, 14]. Figure 1 shows a typical diagram of a neuron.

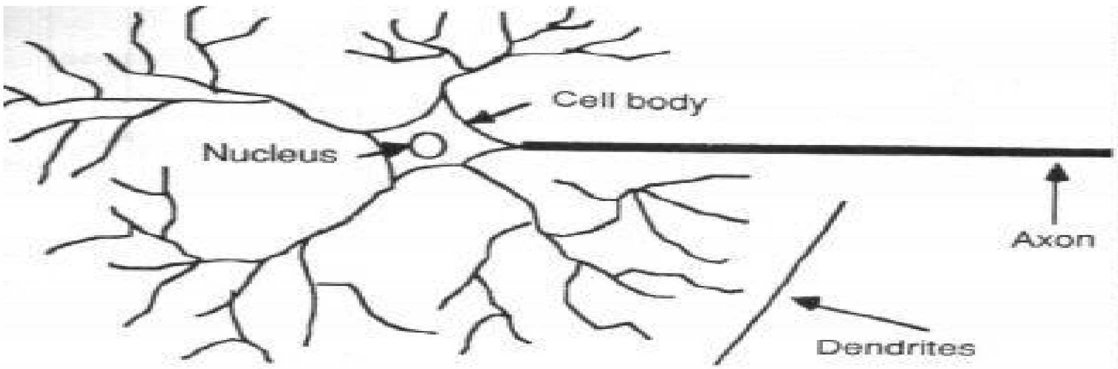

Figure 1: Diagram of a Biological Neuron.

The human brain, for example, contains approximately 1011 neurons, each connected on average to 10, 000 other neurons, making a total of 1,000,000,000,000,000 = 1015 synaptic connections. Neural networks represent an attempt at a very basic level to imitate the type of nonlinear learning that occurs in the networks of neurons found in nature. McCulloch and Pitts [15] are generally recognized as the designer of the first neural network.

They combined many simple processing units together that could lead to an overall increase in computational power. They suggested many ideas like: a neuron has a threshold level and once that level is reached the neuron fires. It is still the fundamental way in which artificial neural network operates. The McCulloch and Pitts's network had a fixed set of weights. The artificial neuron model was introduced by McCulloch-Pitts and it is known as Threshold Logic Unit. The diagram of an artificial neuron is shown in figure 2.

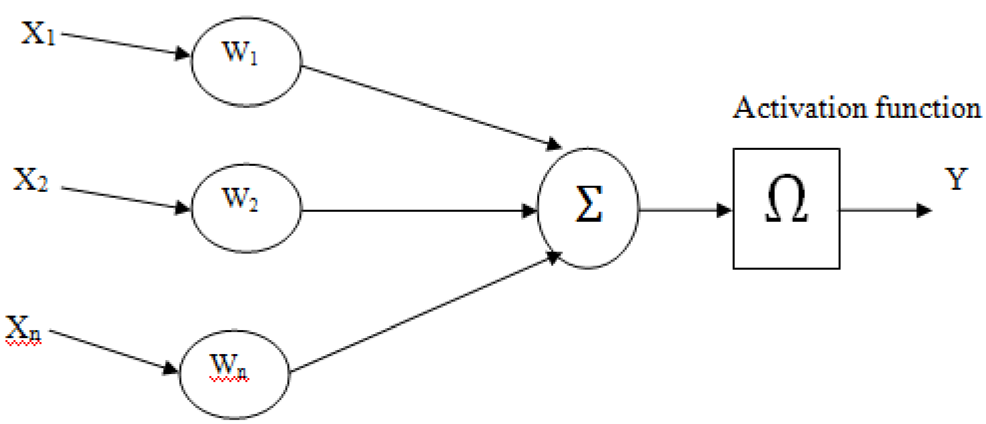

Figure 2: Diagram of an Artificial neuron.

Vol 7. No. 3 - September, 2014
African Journal of Computing & ICT

www.ajocict.net

3. METHODOLOGY

This involves the specification of procedures for collecting and analyzing data necessary to define or solve the problem. The scope of this paper covers the technique used to detect forged signature and type of classification approach that can be used to solve the problem of detecting forged signature is neural network. According to figure 2, a set of input connections brings in activations from other neurons. Then a processing unit sums the inputs, and then applies a non-linear activation function and the output line transmits the result to other neurons. The values $w_{1,}$ $w_{2,}$..... w_n are weights to determine the strength of input vector $x = [x_{1,}\ x_{2,}.......x^T]$. Each input is multiplied by the associated weight of the neuron connection x^T w. the positive weight excites and the negative weight inhibits the node output. The equation can be given thus:

$$I = x^T.w = x_1w_1 + x_2w_2 + \ldots.x_nw_n = \sum_{i=0}^{n} xiwi \quad (1)$$

The node's internal threshold Ω is the magnitude offset.
It affects the activation of the node output Y as:

$$Y = f(I) = f\{ \sum_{i=0}^{n} xiwi - \Omega_k \} \quad (2)$$

To generate the final output Y, the sum is passed on to a non-linear filter f called the activation function which relates the output Y . Neuron generally do not produce an output unless their total output goes above a threshold value. The total input for each neuron is the sum of the weighted inputs to the neuron minus its threshold value. The is then passed through the sigmoid function. The equation for a transition in neuron is :

$$a = 1/(1 + \exp(-x)) \quad (3)$$

Where

$$x = \sum_i a_i wi – Q$$

a is the activation for the neuron
a_i is the activation for neuron $_i$
w_i is the weight
Q is the threshold subtracted.

The activation function f performs a mathematical operation on the signal output. The most common activation functions are; linear function, piecewise linear function, tangent hyperbolic function, threshold function, sigmoid (S shaped) function. Computer understands and process data in binary that is zeros and ones. The methodology used discusses the pre-processing performed, the signature database, and the neural network features. Pre-processing in neural network offers the following pre-processing method and they are background elimination, width normalization, and thinning.

Signature database: since the application is a standalone application, it does not require database but it uses the traditional form of database system which is the file system containing all the scanned signatures and they are encrypted in such a way that only the application can access it for security measures. The application is designed to recognize signature in which the signatures are taken in four places for system to train them and recognize it after which it is train by the because that is the main essence why neural network is used because it can learn and uses what it learnt to do comparison with the new signature that would presented and verification is done by comparing. Training is the act of presenting the networks with some sample data and modifying the weight to better approximate the desired function and that is how the weight comes in.

The weight in a neural network is the most important factor in determining its function. There are two main types of training and they are [1]:

1. Supervised Training: In this type of training, it supplies the neural network with inputs and the desired outputs and response of the network to the input is measured. Also the weights are modified to reduce the difference between the actual and desired outputs.
2. Unsupervised Training: It only supplies the inputs then the neural network adjusts its own weight so that similar inputs cause similar outputs. The network identifies the patterns and differences in the inputs without any external assistance.

3.1 The Preprocessing Step

The system will take in signature by either scanning it in to the system or written on a stylus, after which pre-processing is done and under pre-processing, three steps will take place in this stage and they are:

a) Background Elimination: Many image processing applications require differentiation of objects from the image background. Thresholding is the most trivial and easily applicable method for this purpose. It is widely used in image segmentation. Threshold technique is used for differentiating the signature pixels from the background pixels. In this type of application, interest is on dark objects on a light background and therefore a threshold value *T* called the brightness threshold is appropriately chosen and applied to image pixels. After the thresholding, the pixels of the signature would be 1 and the other pixels which belong to the background would be 0.
b) Width Normalization: Irregularities in the image scanning and capturing process may cause signature dimensions to vary. Furthermore, height and width of signatures vary from person to person and sometimes even the same person may use different size signatures. First there is the need to eliminate the size differences and obtain a standard signature size for all signatures. During the normalization process, the aspect ratio between the width and height of a signature is kept intact and after the

process, all the signatures will have the same dimension. For example width normalization is calculated by:

$$Normalized\ area = \frac{\text{Signature Area}}{\text{Area enclosed in bounding box}}$$

c) Thinning: The goal of thinning is to eliminate the thickness differences of pen by making the image one pixel thick. The comparison is done based on what the system learnt from training. The signature presented is flagged genuine with percentage if and only if the signature is within the range of the already trained signature existing in the system. But if the signature presented is not within the range of the existing signature, it is flagged forge with the percentage of its forgery. The implementation of the system can be achieved with Java program alongside with Netbeans which will serve as editor, compiler and it will also allow the program modules to run at the same time and will make it to be a standalone application. Basically, the design of the system is divided into two stages:

A. Training stage: A training stage consist of four major steps:
1. Retrieval of a signature image from a database (file system)
2. Image pre-processing
3. Feature extraction
4. Neural network training

B. Testing stage: A testing stage consists of five major steps :
1. Retrieval of a signature to be tested from a database
2. Image pre-processing
3. Feature extraction
4. Application of extracted features to a trained neural network
5. Outputting in percentage.

4. RESULTS

Below are a few screen shots or stages of the application consisting of two separate verified signatures with different percentage of authenticity. Interfaces or stages highlighted include the User interface, the Training stage, the verification stage and the Testing stage with fake signature. All these results make up the workings of the Signature Verification System.

4.1 User Interface

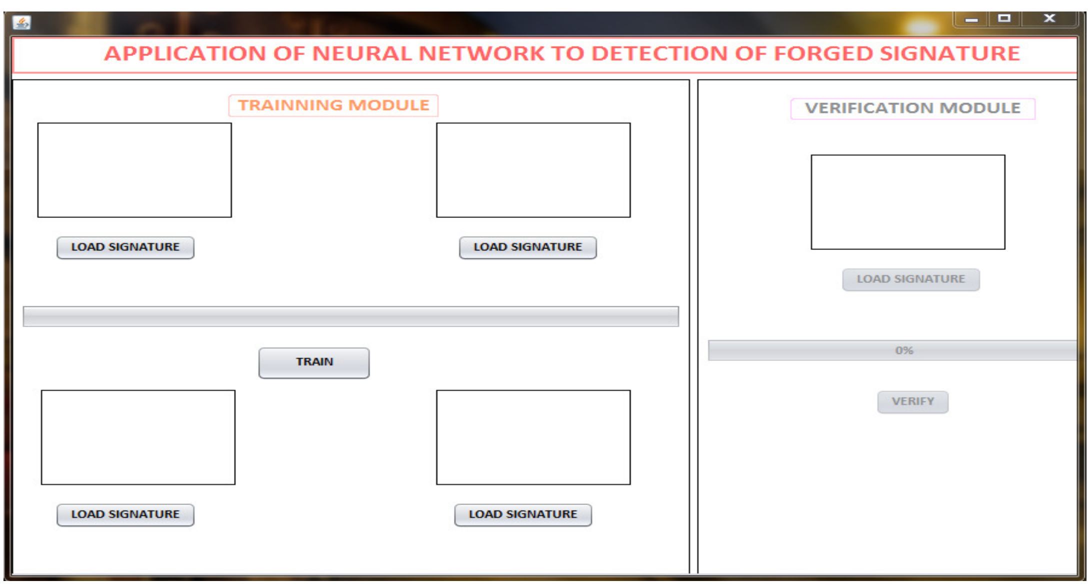

Figure 3: User Interface

African Journal of Computing & ICT

www.ajocict.net

The User Interface is shown in figure 3. This is the first page the user of the application sees. It features an interactive map of the signature detection application.

4.2 Training Stage of the First Sample Signatures

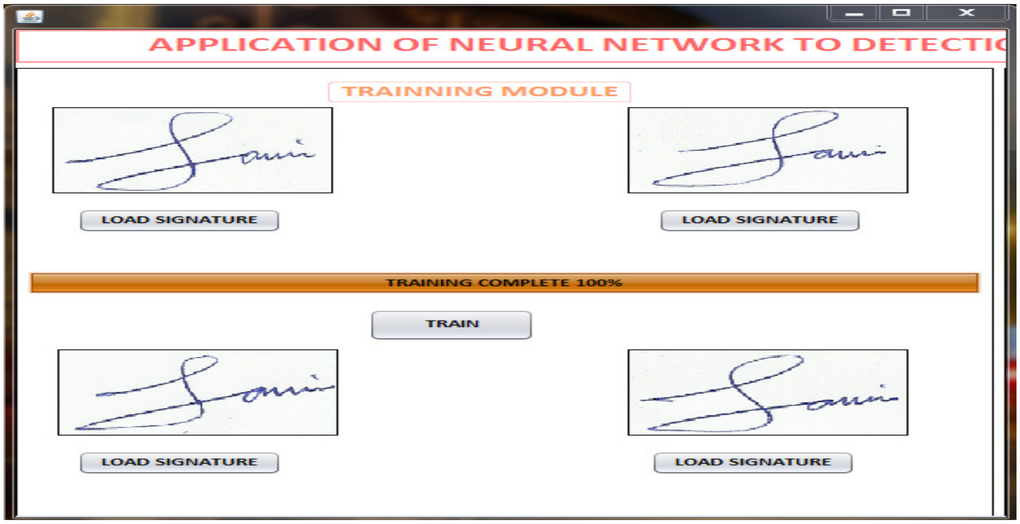

Figure 4: Training Stage

The Training stage is the module where signatures are uploaded into four places and trained so that the application can recognize it, convert the signatures to binary and store them into a file as shown in figure 4.

4.3 Verification Stage of the First Sample Signatures

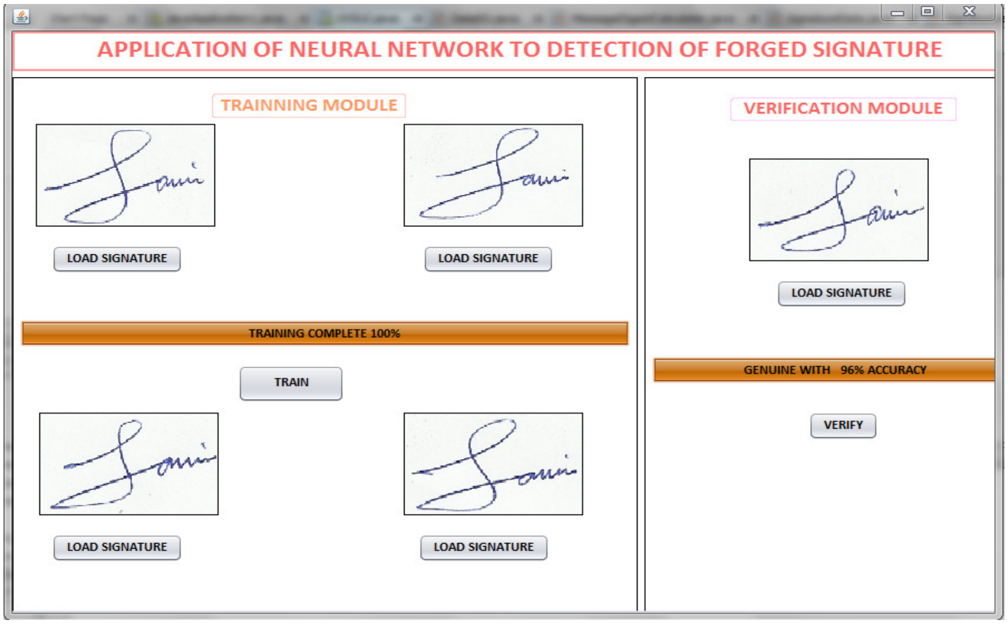

Figure 5: Verification Stage

This stage verifies the signature trained by the application and comparison is made between the signature presented and the already existing ones in the database. This now tells the accuracy of the signature. The signature in figure 5 shows 96% accuracy which is considered genuine for authenticating the user.

4.4 Training Stage of the Second Sample Signatures

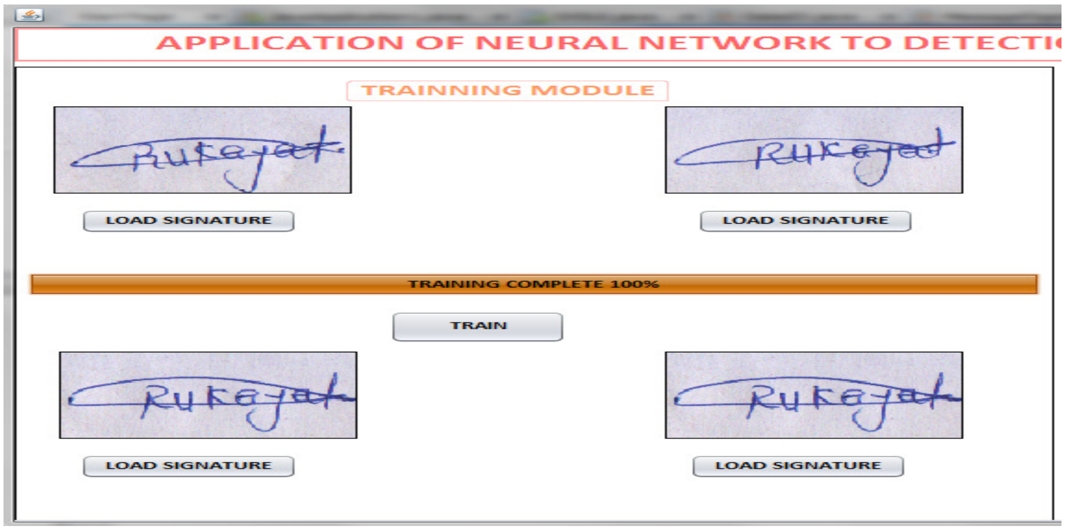

Figure 6: Training Stage

This is also the module where signatures are uploaded in four places and trained so that the application can recognize it, convert the signatures to binary and store them into a file as shown in figure 6.

4.5 Verification Stage of the Second Sample Signatures

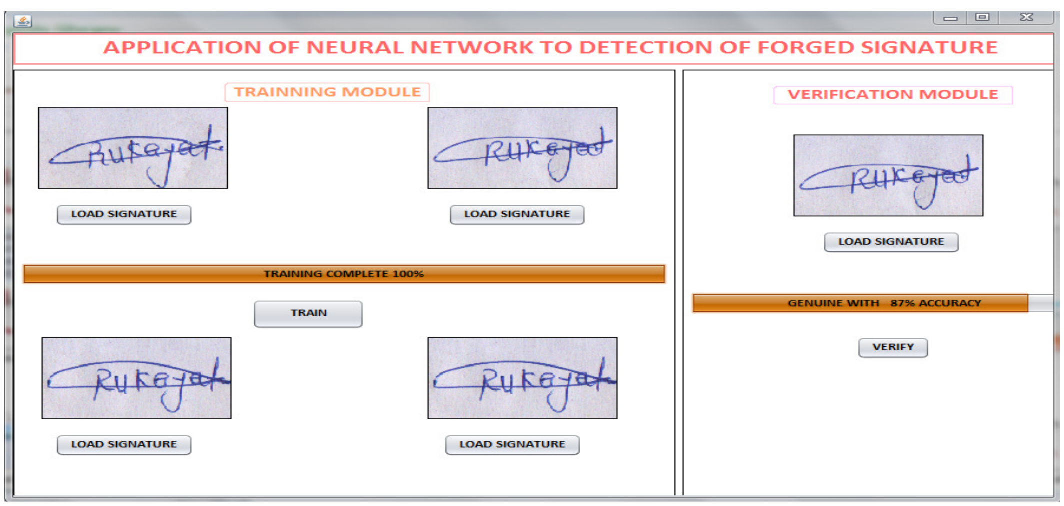

Figure 7: Verification Stage

This stage verifies the signature trained by the application and comparison is made between the signature presented and the already existing ones. This now tells the accuracy of the signature. The verification module indicates 87% similarity between the signed signature and the one in the database as shown in figure 7. This can also be considered for authenticating the user.

5. DISCUSSION OF RESULTS

Figures 4 and 6 shows the training stage for the application to learn the pattern of the signature and save them into database for comparison during verification. It can be seen that the training percentage will reach hundred and these denote completion of training after which verification can later be done by the application. Figure 5 denotes the verification of the first signature, it is seen that the percentage of the signature does not reach 100%, the reason being that the owner of the signature can never sign the exact way he or she has signed before and this has led to the variation in the percentage of the accuracy of the signature. The signature in the verification module can only give 100% if and only if the signature to be verified was taken from the signature used for training the application. In the case of figure 7 which is the verification stage of the second signature, the accuracy of the signature also did not reach 100% because of little variation when the owner is signing such as the speed, pen used, state the owner is when signing and so on. It can be concluded that, the accuracy of the signature can never be 100%.

6. CONCLUSION

In this paper, a signature verification system was been developed by applying neural network. The efficacy of the system was tested on a large database of signatures. The designed system can be used as an effective signature verification system. In achieving this, neural network is used and it is implemented with Java. This paper presents a method for Forged Signature Detection. The extracted features are used to train a neural network by using the back propagation training algorithm. The network could classify all genuine and forged signatures correctly. Technologies used in implementing the signature verification application include the Unified Modeling Language (UML), Java Virtual Machine (JVR), NetBeans compiler and JQuery.

REFERENCES

[1] Abikoye, O. C., Mabayoje, M. A., & Ajibade, R. (2011). Offline Signature Recognition and Verification using Neural Network. International Journal of Computer Applications. Vol. 35, No. 2, 0975-8887.

[2] Bhatia, M. (2013). Off-line Hand Written Signature Verification using Neural Network. International Journal of Application Innovation in Engineering Management (IJAIEM). Vol. 2 Issue 5. ISSN 2319-4847.

[3] OZ, C., Ercal, F. & Demir, Z. (2003). Signature Recognition and Verification with ANN. Third International Conference on Electrical Electronics Engineering, Bursa. Turkey.

[4] Perrig, A. (2001). The Biba One-Time Signature and Broadcast Authentication Protocol. Berkeley, University of California.

[5] Ihler, A. T., Fisher, J. W. & Willsky, A. S. (2001). Nonparametric Estimators for Online Signature Authentication. International Conference on Acoustics, Speech, and Signal Processing, USA. Vol. 6, pp. 3473-3476.

[6] Schimke, S., Vielhauer, C., & Dittmann, J. (2004). Using Adapted Levenshtein Distance For On- Line Signature Authentication. 17th International Conference on Pattern Recognition (ICPR'04), Vol. 2, pp.931-934.

[7] Pivonka, P. and Nepevny, P. (2005). Generalized Predictive Control with Adaptive Model Based on Neural Networks. Proceedings of the 6th WSEAS International Conference on NEURAL NETWORKS, Lisbon, Portugal,16-18, pp. 1-4.

[8] Plamondon, R., & Lorette, G. (1989). Automatic Signature Verification and Writer Identification: the State of the Art 1989-1993. Pattern Recognition. Vol. 22, No. 2, pp. 107-131.

[9] Papadimitrioy, S., & Terzidis, K. (2007). Classification Process Analysis of Bioinformatics Data with a Support Vector Fuzzy Inference System", Proceedings of the 8th WSEAS International Conference on Neural Networks, Vancouver, British Columbia, Canada, pp. 90-95.

[10] Unluturk, M. S., Oguz, K. & Atay, C. (2009). Emotion Recognition Using Neural Networks. Proceedings of the 10th WSEAS International Conference on NEURAL NETWORKS, Prague, Czech Republic, pp. 82-85.

[11] Dule, E., Gokmen, M., & Beratoglu, S. (2010). A Convenient Feature Vector Construction for Vehicle Color Recognition, Proceedings of the 11th WSEAS International Conference on NEURAL NETWORKS (NN '10), G. Enescu University, Iasi, Romania, pp. 250-255.

African Journal of Computing & ICT

www.ajocict.net

[12] Kalenova, D. (2005). Personal Authentication using Signature Recognition. www2.it.lut.fi/kurssit/03-04/010970000/seminars/Kalenova.pdf

[13] Aykanat C. et. al ,(Eds). 2004. Proceedings of the 19th International Symposium on Computer and Information Sciences, ISCIS 2004. Springer-Verlag Berlin Heidelberg New York. pp. 373-380.

[14] Stergiou, C., & Siganos, D. (2003). Neural Networks Retrieved April 1, 2011 from www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs#/report.html.

[15] McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 - 133.

[16] Baradi, V. A. & Kakere, H. B. (2010). Offline Signature Recognition System. International Journal of Computer Science, 48-56.

[17] Bence, K., Benedek, T. & Hassan, C. (2009). Classification Approaches in Offline handwritten Signature Verification. WDET Transaction on Mathematics, VIII(9), 500-509.

[18] Blumenstein, M., Armand, S. & Muthukkumarasamy, (2006). Off-line Signature Verification using the Enhanced Modified Direction Feature and Neural Based Classification. International Joint Conference on Neural Networks.

[19] Dolfing, J. G., Aarts, E. H. & Van Oosterhout, J. J. (1998). On-line Signature Verification with Hidden Markov Models. Proceedings of the 14th International Conference on Pattern Recognition, Brisbane. 1309-1312.

[20] Enturk, T. S., Zgunduz, E. O., & Karshgil, E. (2005). Handwitten Signature Verification Using Image Invariant and Dynamic Features. Proceedings of the 13th European Signal Processing Conference (EUISIPCO), Antalya.

[21] Fausett, L. (1994). Fundamentals of Neural Networks. New York: Prentice Hall Publishing.

[22] Gallinari, P., Thiria, F., Badran, F., & Fogelman-Soulie, F. (1991). On the Relation between Discriminant Analysis and Multilayer Perceptron, Neural Networks. IV, 349-360.

[23] Geok, S. N., & Ong, H. S. (1993), A Neural Network Approach for Offline Signature Verification. IEEE TENCON.

[24] Gupta, G. K. (2006). The State of the art in On-line Handwritten Signature Verification. Monash University, Faculty of Information Technology, Australia.

[25] Jain, A., Griess, F. S. C. (2010). On-line Signature Verification Pattern Recognition, WSEAS Transaction on Mathematics.

[26] Manoj, K. (2012). Signature Verification Using neural Network. International Journal on Computer Science and Engineering (IJCSE). 1498-1504.

[27] McCabe, A., & Trevathan, J. (2007). Implementation and Analysis of a Handwritten Signature Verification Technique. Pp 48-58.

[28] McCabe, A., Trevathan, J. & Wayne, R. (2008). Neural Network-based Handwritten Signature Verification. Journal of Computer, School of Mathematics, Physics and Information Technology, III(8).

[29] Parker, J. R. (2005). Simple Distance between Handwritten Signatures. University of Calgary, Department of Computer Science.

[30] Plamondon, R. (1995). The Handwritten signature as a Biometric Identifier: Psychophysical Model and System Design. IEEE Conference Publication, pp. 23-27.

[31] Prashanth, C. R., Raja, K. B., Venogopal, K. R., & Patnaik, L. M. (2009). Standard Scores Correlation Offline Signature Verification System. International Conference on Advances in Computing, Control and Telecommunication Technologies.

[32] Quen-Zong, W., I-Chang, J. & Suh-Yin, L. (1997). On-line Signature Verification Using LPC Cepsrum and Neural Networks. IEEE Transactions on System, Man, and Cybernetics, pp. 148-153.

[33] Rajasekaran, S., & Vijayalakshmi, G. A. (2006). Neural Networks, Fuzzy Logic, and Genetic Algorithms", Prentice-Hall India Pvt., Sixth printing, May 2006. R. J. Vidmar (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. IEEE Trans. Plasma Sci. [Online]. 21(3). pp. 876—880.

[34] Sandeep, P., & Shailendra, D. (2011). Neural Network-based Offline Handwritten Signature Verification System using Hu's Moment Invariant Analysis. International Journal of Engineering and Advanced Technology (IJEAT) I(I), 2249-8958.

[35] Srihari, S. N., Xu, A. & Kalera, M. K. (2004). Learning Strategies and Classification Methods for Off-line Signature Verification. Proceedings of the 9th International Workshop on Frontiers in Handwritten Recognition (IWFHR-9).

[36] Sureerathanan, S., Huynh, N. P., Sureerathanan, N. & Nlkos, M. E. (2006). The Optimal Multi-layer Structure of Back Propagation Networks. Proceedings of the 7th WSEAS International Conference on Neural Networks, pp. 108-113. Cavat.

[37] Xiao, X., & Leedham, G. (1999). Signature Verification by Neural Networks with Selective Attention. Journal of Applied Intelligence, XI(2), 213-223.

[38] Zois, E., & Anastassopoulos, V. (1996). Methods for Writer Identification. Proceedings of the Third IEEE International Conference on Electronics, circuits and Systems (ICECS).

[39] Bhattacharyya, D., & Kim, T. (2010). Design of Artificial Neural Network for Handwritten Signature Recognition. International Journal of Computers and Communications. Vol. 4, Issue 3.

[40] Plamondon, R., & Leclere, F. (1994). Automatic Signature Verification: the State of the Art 1989-1993. International Journal of Recognition and Artificial Intelligence. Vol. 8, No. 3, pp. 643-660.

[41] Verma, R. & Rao, D. S. (2013). Offline Signature Verification and Identification using Angle Feature and Pixel Density Feature and both Method Together. International Journal of Soft Computing and Engineering (IJSCE). ISSN: 2231-2307, Vol. 2, Issue 4.

[42] Bhattacharyya, D., & Kim, T. ().Signature Recognition using Artificial Neural Network. Advances in Computational Intelligence, Man-Machine and Cybernetics. Pp. 183-187.

[43] Yusof, M. H. M., & Madasu V. K. (2003), Signature Verification and Forgery Detection System. Student Conference on Research Development (SCOReD) Proceeding, Purrajaya, Malaysia.