NEURAL NETWORK MODEL FOR DETECTION OF RESULT ANOMALIES IN HIGHER EDUCATION

Ziweritin S.[1], Baridam B. B.[2] and Okengwu U. A.[3]

[1]Department of Estate Management and Valuation, Akanu Ibiam Federal Polytechnic, Unwana-Afikpo, Ebonyi State, Nigeria.

[2,3]Department of ComputerUniversity of Port-Harcourt Choba, Rivers State, Nigeria

stanlo4godsluv@yahoo.com,barilee.baridam@uniport.edu.ng, ugochi.okengwu@uniport.edu.ng

Received: 03-08-2020

Accepted: 20-08-2020

ABSTRACT

The performance of students in tertiary institutions within and outside Nigeria is based entirely on end of academic-session examination. The processes involved in carrying out semester examinations are complex and crucial in tertiary institutions and confidentiality must be ensured. Anomaly detection in results is an important and increasing problem that has been well-studied within diverse research areas and application domains. The admission of students into different departments of tertiary institutions in Nigeria is increasing at a very high rate and has now reached a position where it is becoming difficult for the available manpower and the existing system to cope with the magnitude of irregularities between the CA and exam scores to detect anomalies, within the given time span leading to delay in approving students semester results for decision making. In this paper, an efficient neural network model is developed to systemically detect anomalies in students' results as an effective measure which can enhance the efficiency and accuracy of the system. The study and the preliminary designs were carried out using the Object Oriented and Design Methodology (OODM), simulated using MATLAB in the design, training and testing aimed at detecting anomalies from the dataset. The system was successfully tested according to the design specification with 96% accuracy level in comparison with existing methods.

Keywords: Neural network, anomaly detection, academic results, object-oriented design, simulation

I. Introduction

Machine learning is a subset of artificial intelligence that uses statistical methods to help machines learn through experience (Talwar & Kumar, 2013), where the experience gained is used to help computers perform specific tasks intelligently, Learning systems can carry out complex processes by learning from data patterns rather than following pre-programmed rules. Within the field of machine learning there have been algorithmic advancements, which provides a better machine learning power. In recent years the concept of machine learning techniques have been used in many applications domain and is a core concept for intelligent systems (Singh, Kumar, & Kaur, 2014). *A*s the field of computing progresses, machine learning has supported potentially transformative advances in different study areas or fields, and the social and economic opportunities that follow are very significant.

Within the Nigerian institutions, processing of students' results across tertiary institutions are largely manual in nature. These results have some form of anomalies in respect to the discrepancy that occurs between students continue assessment(CA) and exam scores. The negative side of this is that it causes delay in computing and result approval by dedicated authorities:- It is also prone to some human errors.

Examinations are one of the most important activities that take place in institutions of learning. In many tertiary institutions of higher learning series of meetings are held at the departmental, faculty and senate levels to manually examine and approve computed student examination results which comprises of assignment, course work, test or quiz score as CA and exam scores. At such board meetings, student results are checked, scrutinized, and then reasonable explanations provided for any anomaly that is detected prior to approval. This paper intends to develop a neural network model that can detect students' result anomalies to increase the integrity of the process.

The paper is divided into sections as followings: Section II presents a brief review of some of the previous approaches to the study area and the gap in exploring the proposed model; Section III, introduces the materials and methods which the different methods adopted and materials used for developing the model; Section IV, focuses on the results and detailed discussion of results; Section V presents the conclusion to the paper.

II. Related Works

Despite the substantial advances made in the study of artificial neural network models in many machine learning problems, there is a relative scarcity of neural network learning approach for exam anomaly detection (Hamza et al.,2016) However, there are some research works that have direct bearing on the work presented in this paper.

Hamza et al. (2016) carried out an overview of different supervised and unsupervised anomaly detection algorithm in the detection of intruders to improve the intrusion detection system(IDS). The results of the experiment reveals that supervised learning methods were significantly better than the unsupervised if the test data contains no unknown attacks. Hamza, Ruqayyah, and Mohammed (2016); developed a decision tree model with labeled leaf nodes containing 42 instances using the Waikato Environment for Knowledge Analysis (WEKA) as a pre-processing tool. Some important parameters of the j48 and C4.5 decision tree algorithms were changed for comparison after repeatedly running few initial experiments. The confidence level for pruning was set to 0.25 and the minimum number of instances per leaf node was fixed at 2. The dataset was randomly divided into 405 or 60% for training and 266 or 40% for testing. The challenges of this process in the use of decision tree for detecting borderline failure anomalies is that, it was unable to identify any of the two anomalous cases during training which results in sensitivity value of zero. Moreover, at testing stage, sensitivity could not be computed because of biasness(systematic error) meaning that the test data was not part of their training dataset resulting in a sensitivity value of zero and there was poor performance due to instances of borderline failure cases in the training dataset.

John et al. (2017) in their work used a futuristic approach of combining K-means and decision tree. They chose K-means clustering thereby accelerating the convergence of clusters on large datasets. The first stage of K-means clustering was performed on training instances to obtain k disjoint clusters based on Andrew et al. (2011) approach. Each k-means cluster represented a region of similar instances or data points in terms of Euclidean distances between the instances and their cluster centroids and the second stage of combining K-means and the decision tree was cascaded with the decision tree learning by building an iterative dichotomiser 3 decision tree using the instances in each input data to K-means cluster to analyze, grouped data into clusters. It has different number of clusters K_1, K_2… K_n and each cluster is having its own rules, for unknown data by default considered to be N (representing the number of clusters) which depends on the criteria given.

Jamal, and Aishwarya (2017), compared the efficiency of artificial neural networks(ANNs) and support vector machine(SVM), with the hope of providing an efficient and more accurate detection system. The mean value of their proposed model was computed from the training cycle. This produced higher performance rate with intrusion detection benchmark dataset for detecting user-to-root(U2R) and remote-to-local(R2L) attacks that affects a larger number of computers in the world. (a). User-to-root attack (U2R): the intruder have access to the target system using normal user account or valid user authentication and attempt to abuse or take advantage of the vulnerabilities of the user system in order to gain access to supper privileges. Attacks belong to the U2R are as follows: buffer overflow attack, load-module and rootkit etc (b). Remote-to-local(R2L) attack: The Attacker in this scenario uses existing machine vulnerabilities to access target account in which the intruder does not own the target system but access it locally. Examples of R2L attacks are as followed: multi-hope, guess and password etc

Finally the result of the k-means and ID3 tree was classified as data into anomaly. The limitations in this work is that; their proposed systems cannot work well with large volume of data and any little variation in data set give rise to a different result. Ruggieri (2002) developed and presented the results of the running time behavour of the E4.5 and C4.5 decision tree models using three different strategies for detecting anomalies which highlighted some efficiency and improvements. The dataset was subdivided into four categories: x_train, y_train, x_test and y_test and applied a function to perform training with entropy that fit into the decision tree to determine the accuracy level of both models. He concluded that the C4.5 tree performed better with small test and training dataset than the E4.5. Hawkins (2001) proposed a simple multi-layer neural network model comprises of two(2) inputs, eight(8) hidden, and 2 output layers but unable to detect specific attacks types centred on computer networks for denial of network services. However, the efficiency and the accuracy of their proposed model controlled by the adjustable epochs, hidden-layers and desired error parameters.

Ruhui (2008); used a novel hybrid-based radial basis function (RBF) in neural network and quantum-behaved particle swarms optimization(QPSO) proposed for network anomaly detection. The results of QPSO produced high detection rate while maintaining a low false positive rate with some adjusted parameters compared to the RBF with lower detection and maintained false positive rate respectively. The quantum-behaved particle swarm optimization (QPSO) algorithm, which performed better than the RBF in neural network and converges faster.

The hybrid algorithm in training RBF neural network was further improved by Fujimaki, Yairi, and Machida (2005) with new evolutionary algorithm, carried out based on the hybrid of quantum-behaved particle swarm optimization (QPSO) by the addition of gradient descent algorithm (GDA), employed to train the radial basis function(RBF) in neural network which increased the intelligency of the RBF model in detecting more attacks. The added GDA helped the RBF to produce a higher detection rate and maintained a low false positive rate compared to their existing model.

III. MATERIALS AND METHODS

It is important to express the fact that efforts have been made in the area of detecting result anomalies. The exact solution is mainly the bases of discussion which we intend to build on the areas that we have identified some lapses. In the architecture of the existing system we were able to identify and narrow our study to CA and Exam type of anomalies in student results with a well define anomalous data boundaries after training the machine using the proposed neural network model. The experimental dataset we intend using in designing the model are promising in terms of accuracy, precision, and time complexity.

In order to know the efficiency of the neural network model over the exact system solution, we intend to develop a model and compare the results with the proposed system as an approximated solution to produce better and more accurate result in avoiding truncated errors for handling problems relating to floating point values recorded as grade point(GP) in detecting result anomalies.

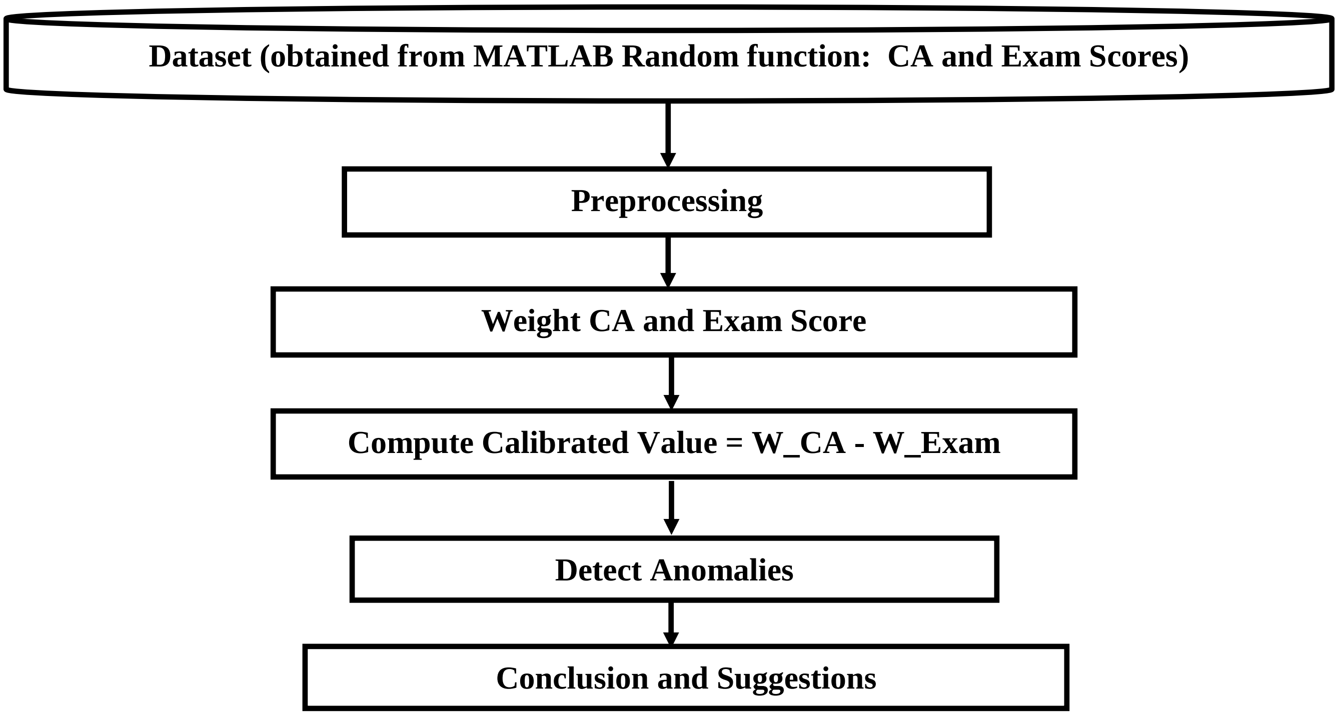

Dataset (obtained from MATLAB Random function: CA and Exam Scores)
Preprocessing
Weight CA and Exam Score
Compute Calibrated Value = W_CA - W_Exam
Detect Anomalies
Conclusion and Suggestions

Figure 1: The Architecture of the old system by Peddabachigari, et al. (2007).

The proposed system is based on the feed-forward artificial neural network (ANN) algorithm for result anomaly detection with a weighted student's CA (WCA) as added advantage to the proposed system for detecting CA and Exam Anomalies. The neural network model is trained with a dataset for prediction using MATLAB while a database is designed for the application where the details of student's such as name, registration number, and courses offered, course codes, course titles, CA's and exam scores are stored and implemented with VB.NET for deployment. Access to the system is through user authentication, where each user is prompted to supply his/her username and password to gain access if it matches in the database to show student's CA and exam scores with the detected anomalies. Otherwise, access will be denied. The score sheets are presented and stored using a data grid view control in the form of Microsoft excels spreadsheet. The user is prompted to enter Student CA's and exam scores into the application through the result icon that pops down first and second semester score sheets. The floating point computations are done by the application from which anomalies will be detected as if found using the CA's and exam scores. The proposed system is developed to meet up with the functional (core system requirements) to detect irregularities found in student results and non-functional requirements using machine learning technique or paradigm (neural network)

The proposed model is developed in a modular fashion with each module performing a specific task.

Algorithm of the Proposed(neural network) Model

Step 1: Initialization of synaptic weights.

1.1 Pre-processing of input block with weighted CA before passing into the ANN

Step 2: Two input block pass through ANN and real output block (**O_r**) is compared to expected output block (**O_e**).

(a): If there is a total match, pattern will be successfully recognized and there is no need to change synaptic weights. If pair is the last one from the training set, then process may eventually end here.

(b): If any mismatch is discovered, Then ANN would need to be trained by learning Mechanism

(c): Define anomaly sensibility – defines the boundaries for data points

Step 3: Specific output parts which are anti-coincident define the required changes to synaptic weights according to δ rule. Weights are adjusted gradually from the output layer to the input layer. This process is entitled as a backpropagation. Algorithm continues with the step 2 until the training set is empty.

Step 4: Record the training set for patterns similar but not exactly that of the training set.

Step 5: If training process is complete; then evaluate the overall faultiness of ANN.

According to evaluation results it is determined whether the network is sufficiently trained or the whole process must be repeated in the new iteration.

a. Design of the proposed system

The design examined the architecture, synthesizes system components, modules, interfaces, and data for the system to satisfy core requirements. It is done in such a way that the system becomes interactive to the user. According to Blanchard *et al.* (2015) system design emphases on data structure, software architecture, procedural details (algorithms etc) and interface between the modules. For the design of the anomaly detection model using neural network, Use-case diagram, and flowchart will be used.

b. Methodology used in the proposed model

In this study, we adopted the Object Oriented Analysis and Design Methodology(OOADM). The object-oriented approach combines data and processes known as methods into single entities called objects. The stages for object–oriented design can be identified as: (a). Definition of the context of the system which could either be static or dynamic. (b). Designing system architecture by partitioning the system into layers and each layer is decomposed to form the subsystems.
(c). Identification of the objects in the system. The objects identified are grouped into classes. (d). Building of design models. (e).Design of object interfaces. The methodology breaks down the components of the proposed system based on the objects that surround the system and using the object components to build the new system around the identified objects. The new system will have relationships, activities and even dependences around the identified objects. Messages for the performances of any identified activity will be delivered to objects which will interact with the corresponding method to make sure that the activities are carried out within the system. When there are existing components outside of the system message can also be delivered to it to perform certain actions and return corresponding result back to the calling methods or classes. In other to achieve a well-organized system, section of activities will be categorized into classes in a way that will make each group of activities and the processes easier to implement by Aneetha, and Bose (2012).

c. Neural Network Model

The neural network evolves in time and operates with each of its input units in parallel, and each unit presented mathematically. A collection of interconnected units changing in time and yet operating in parallel can be referred to as an Artificial Neural Network(ANN) by Aderemi and Andronicus (2017); Axons and Dendrites are represented by the units while a weight μ_{ij} that moderates the influence of unit j on unit i are assigned to each connection (j,i). We can now say that the ANN is a weight-directed graph where each node i is linked with a threshold value of s_i along with a transfer function f_i, in such a way that unit i will deliver an output y_i of the form: $y_i = f_i(\sum \mu_{ij} x_j - s_i)$, where the jth input of this unit is x_j and the sum of all its weighted CA inputs is $\mu_{ij} x_j$. For producing the output y_i unit i is activated only if this sum is greater than the threshold s_i, if not, then unit *i* is considered to be in an inactive state. To produce some desired behaviour on the neural network, the parameters μ_{ij} and s_i can be adequately adjusted. To alter the expected results to meet the new requirements made to the weighted CA's and bias parameters in order to train the neural network can be trained by adjusting the weighted CA by adding input bias and output bias.

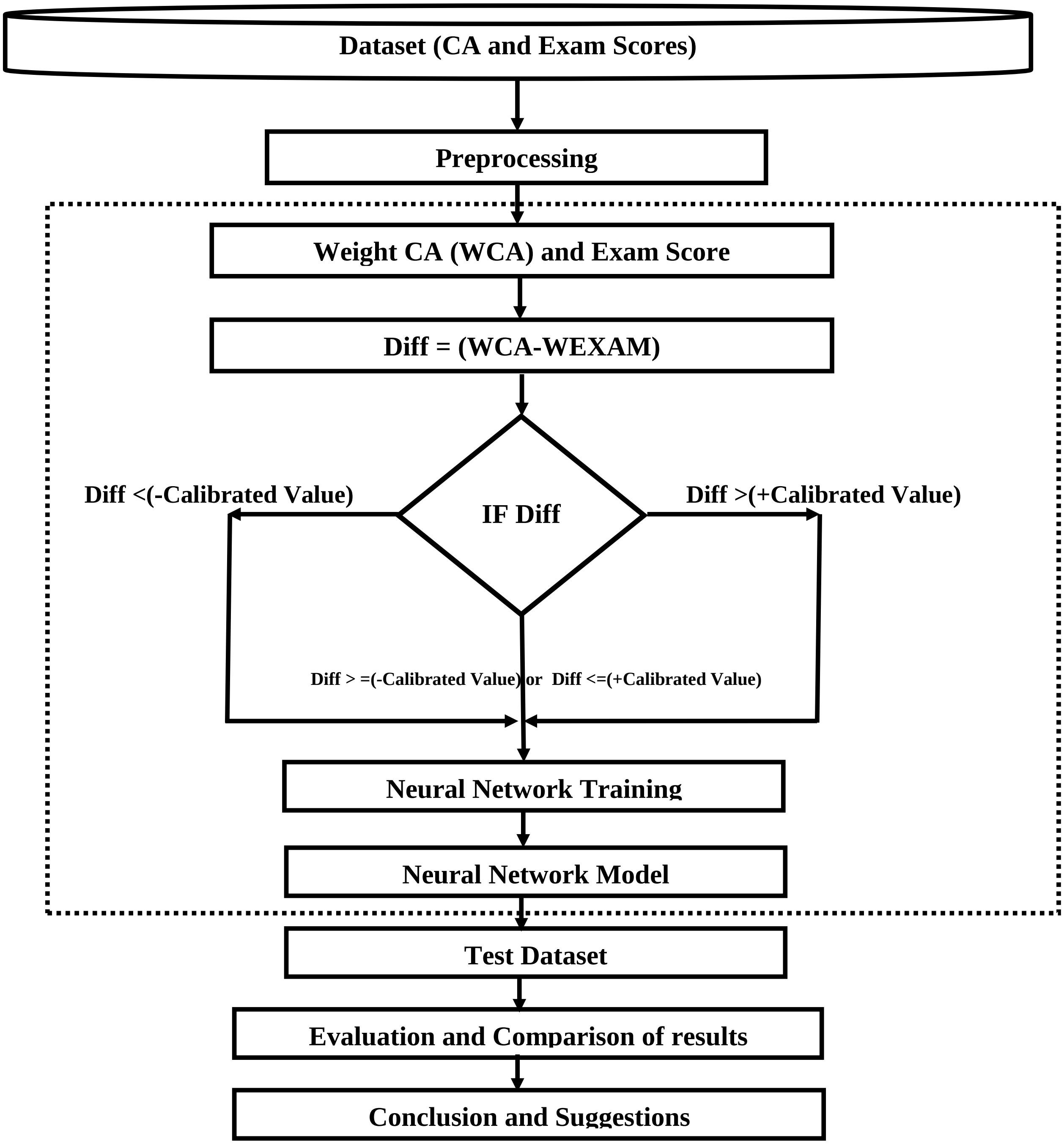

Dataset (CA and Exam Scores)
Preprocessing
Weight CA (WCA) and Exam Score
Diff = (WCA-WEXAM)
IF Diff
Diff <(-Calibrated Value)
Diff >(+Calibrated Value)
Diff > =(-Calibrated Value) or Diff <=(+Calibrated Value)
Neural Network Training
Neural Network Model
Test Dataset
Evaluation and Comparison of results
Conclusion and Suggestions

Figure 2: The Architecture of the proposed System(Source: Alireza & Bita (2008).

d. Artificial Neural Network model for the proposed system

The neural network architecture of the proposed system is made up of two(2) input variables(CA and exam scores), a hidden layer and the(2) output layers(CA and exam anomalies) with a weighted CA(WCA) value to standardize abnormal data points that falls within the rejected region and normal data points that falls within the accepted region by a calibrated value which is the accepted level of anomaly been computed. The design and architecture of ANN selected for the result of anomaly detection system is based on the feed - forward neural network shown in figure 2 bellow as sample of neural network architecture.

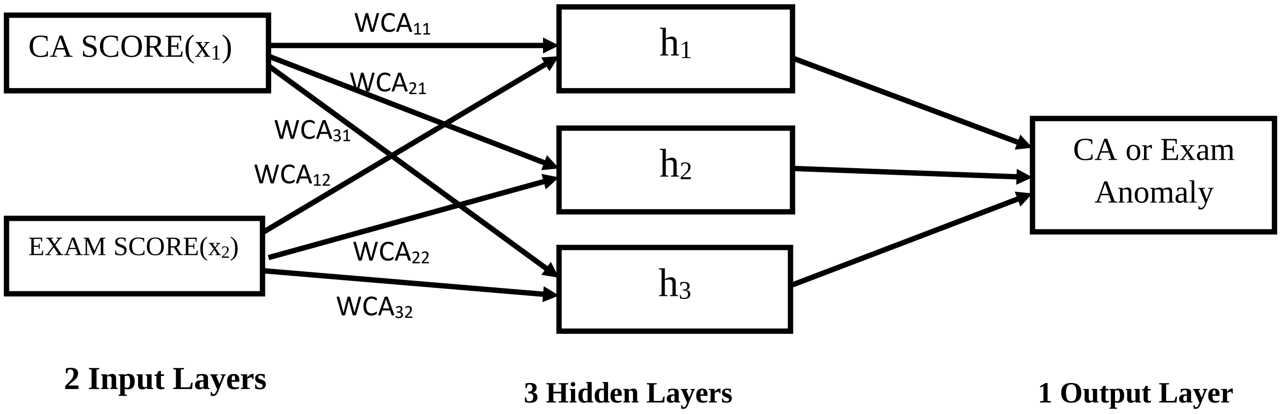


Figure 3:Neural Network representation of selected result anomalies(**Source:** Alireza, and Bita, 2008)

The pieces that remain in between the input and the output layer of the neural network model are the hidden layers in neural network. The increase in hidden layer does not affect the accuracy and efficiency of the model however depends on the complexity of the problem and its tolerant level but decreases neural network training time. The output of the input will be stored in the hidden layer and that becomes the input of the hidden layer. The output of the hidden layer becomes the input of the output layer as shown in figure 2. The weighted matrix of the feed-forward neural network model shown in the above figure 2 becomes.

$$\begin{pmatrix} WCA_{11} & WCA_{12} \\ WCA_{21} & WCA_{22} \\ WCA_{31} & WCA_{32} \end{pmatrix} \tag{1}$$

$$h_i=WCA_{ij}*I_i \tag{2}$$

$$\begin{matrix} h_1 & ------\rightarrow \\ h_2 & ------\rightarrow \\ h_3 & ------\rightarrow \end{matrix} \begin{pmatrix} WCA_{11} & WCA_{12} \\ WCA_{21} & WCA_{22} \\ WCA_{31} & WCA_{32} \end{pmatrix} \text{X} \begin{bmatrix} X_1 \\ \\ X_2 \end{bmatrix} \tag{3}$$

The matrix product of the weighted sum becomes

$$\begin{array}{ll} h_1 & ------\rightarrow \\ h_2 & ------\rightarrow \\ h_3 & ------\rightarrow \end{array} \begin{pmatrix} WCA_{11} * X_1 & + & WCA_{12} * X_2 \\ WCA_{21} * X_1 & + & WCA_{22} * X_2 \\ WCA_{31} * X_1 & + & WCA_{32} * X_2 \end{pmatrix} \quad (4)$$

The matrix product is the product of the weighted matrix and the column of the two input(CA(x1) and Exam Score(x2)) where $h_1, h_2,$ and h_3 are the hidden layers that stores the results of the first phase and $X_1$and X_2 are the input variables as a single column.

$$h_i=WCA_{ij}*B_i \quad (5)$$

Where B_i is the bias added to the result of the hidden layer

$$\begin{pmatrix} WCA_{11} * X_1 & + & WCA_{12} * X_2 \\ WCA_{21} * X_1 & + & WCA_{22} * X_2 \\ WCA_{31} * X_1 & + & WCA_{32} * X_2 \end{pmatrix} + \begin{bmatrix} b_1 \\ \\ b_2 \end{bmatrix} \quad (6)$$

$$h_i=\sigma(WCA_{ij}*I_i +B_i) \quad (7)$$

Where σ is the sigmoid function

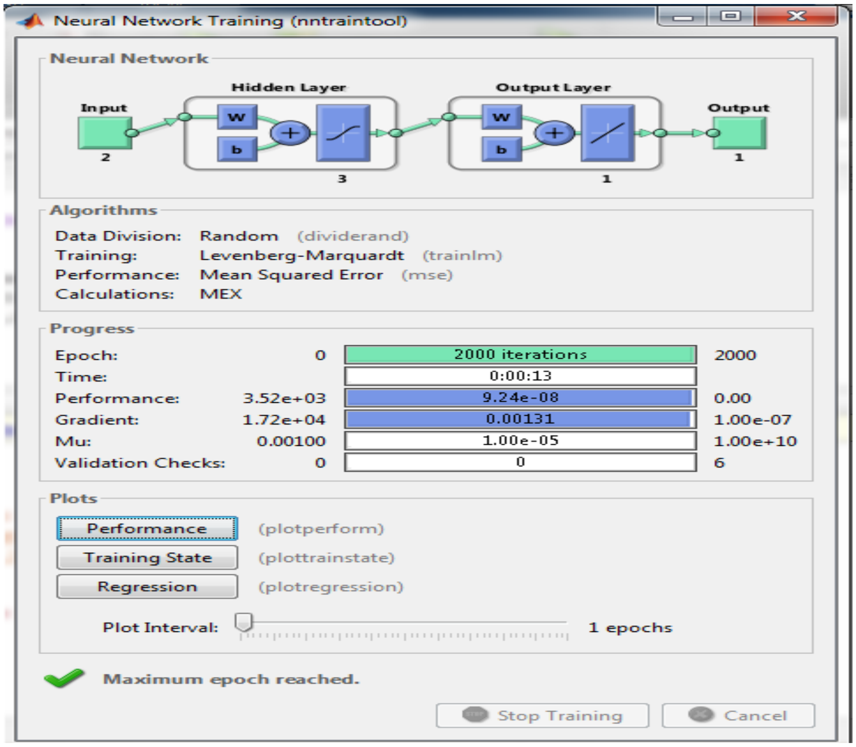
$$Output(O)= \sigma(WCA_{ij}*I_i +B^o) \quad (8)$$

IV. RESULTS AND DISCUSSION

The discussion about the methods adopted and the results obtained are here presented.

a. The dataset

The data used in the model is sourced from an experimental dataset generated randomly in MATLAB using the randomized function consist of instances that gives detailed information of student's weighted CA and Exam scores, and every data that flow through the input layers of the neural network model has different weighted CA and Exam scores up to 1300.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

```
Epoch:                      0   2000 iterations   2000
Time:                               0:00:13
Performance:         3.52e+03      9.24e-08       0.00
Gradient:            1.72e+04      0.00131        1.00e-07
Mu:                   0.00100      1.00e-05       1.00e+10
Validation Checks:          0         0           6
```

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Figure 4: The MATLAB IDE showing Graphic user interface

Figure 4 shows the MATLAB graphic user interface of the neural network with two(2) input layer that accepts weighted CA and exam scores, three hidden layers, one output layer with a progressive bar indicating the number of iterations (Epoch) that runs from 0 to 2000 iterations, time, performance, Gradient, Mu and validation check respectively at the time of training the network model.

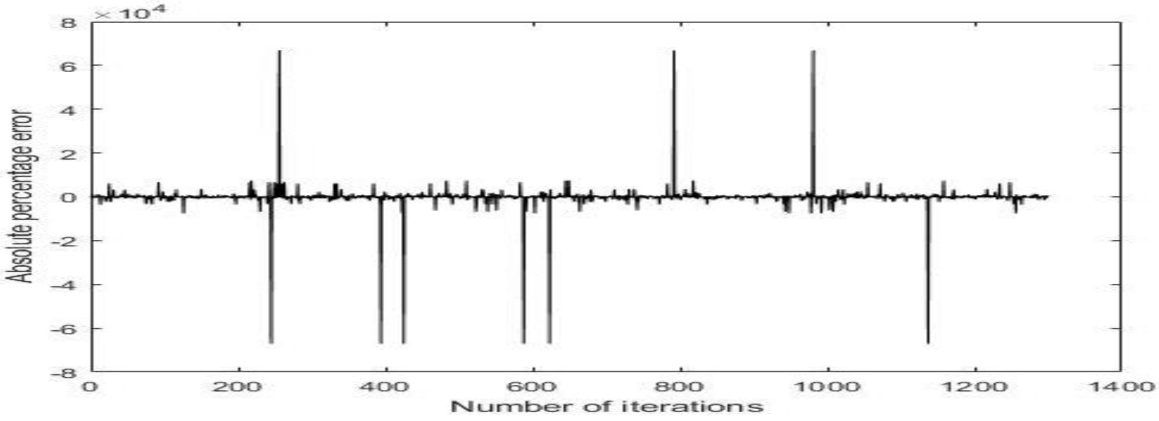


Figure 5: Graph of absolute Percentage error

Figure 5 shows the graph of vertical axis (absolute percentage error) against the horizontal axis (number of iterations) from the figure 4 displaying the convergences of different data points; just few points falls far and out from the points of convergences(zero point or plane) during training stage shown in the above chart.

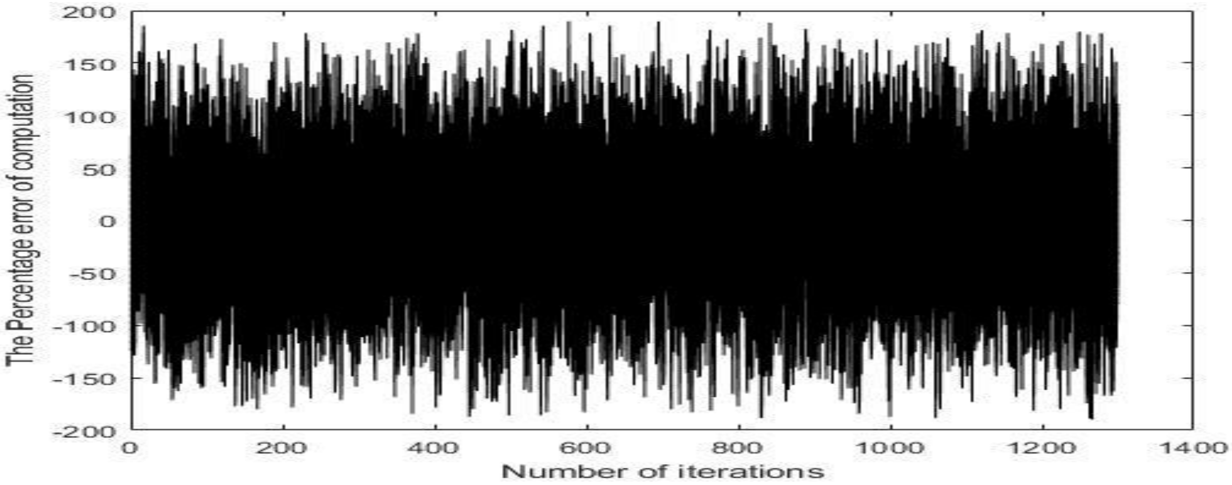


Figure 6: The graph of percentage error in computation

Figure 6 shows the neural network graph of percentage error of computation represented at the vertical axis against the number of iterations represented at the horizontal axis. The density of converging points ranges fall within 100, 50, 0, -50 and -100 at the vertical axis(the percentage error of computation) while the horizontal axis representing the number of epochs or iterations ranges from 0 to 1,400 but the density of points clusters within 0 to 1,300 with some few point falling outside the required margin (accuracy level) of computation greater than +100, 150 and 200 or less than -100, -150 to -200 at the horizontal axis.

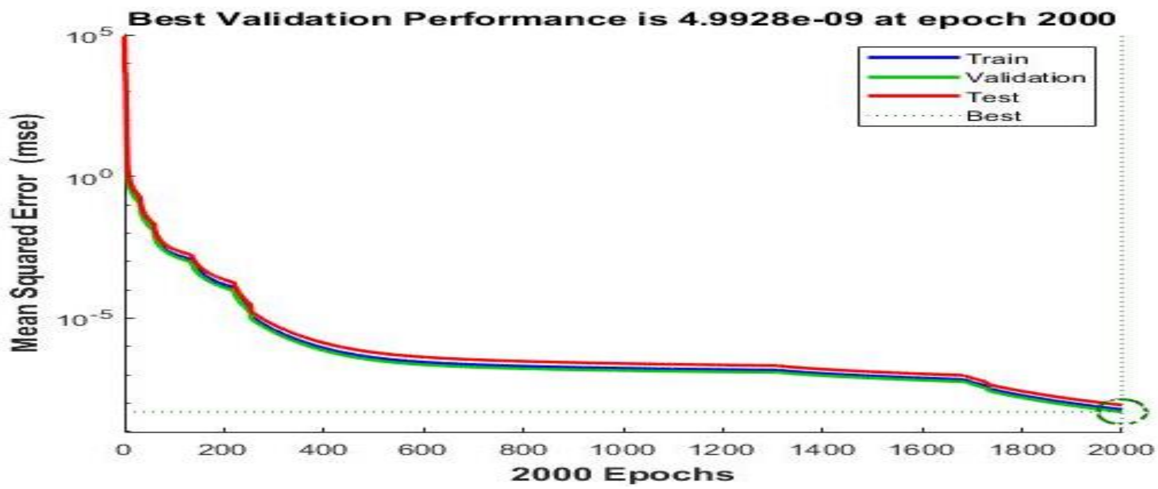


Figure 7: Neural Network Training Performance plot

figure 7 shows the performance from a confusion matrix demonstrating the decreases in errors of training, validation, testing stages and terminates at its tolerance level within the iterations to match the best possible solution in terms of accuracy for the final trained neural network model for detection of CA and exam anomalies which demonstrates the training, validation and testing errors all decreases until iteration 2000 to match the best solution. From the test it does not appear that any over-fitting and under-fitting has occurred, since testing, validation and training errors decreases before iteration 2000 and produced the target exactly at 2000 epochs respectively. The test coincide with the best solution at exactly 2000 epochs or iterations been circled with green colored circle in the legend which falls within its tolerance level as valid test shown in figure 6 above. The machine was able to learn and produced the exact solution with high accuracy level at 2 place of decimal with best validation performance value of 4.9928e-09 at epoch 2000.

Table 1: The Processed Data

S/N	WCA	CA	WEXAM	EX	ADFF	MDFF	ANOMALY
1	3	1	31	22	28	28.428	CA_anomaly
2	27	8	19	13	8	8.4284	Exam_anomaly
3	57	17	7	5	50	49.857	Exam_anomaly
4	30	9	61	43	31	31.428	CA_anomaly
5	67	20	100	70	33	33	CA_anomaly
...	....	...	...	...	...	...	...
2299	13	4	31	22	18	18.429	CA_anomaly
2300	70	21	86	60	16	15.714	CA_anomaly

By comparing what the machine leant and produced as the difference gotten from the weighted CA(W_CA) and the exam(W_Exam) scores as the predicted difference (PDFF) and the exact or actual difference(ADIFF or Exact difference) between the existing and new system revealed that; the proposed model produced the same result as the exact solution in two places of decimal(2pd) in terms of accuracy and precision. In the first row of the of figure 5 displayed 28 as the exact value gotten from the difference between the weighted CA and exam and 28.428 as the neural network predicted value with an error value of 0.428, row two 8 as the exact difference and 8.4284 as the predicted value with an error value 0.4284 to produce the exact value at two place of decimal, third row 50 as the exact value and 49.857 as the predicted neural network value etc.

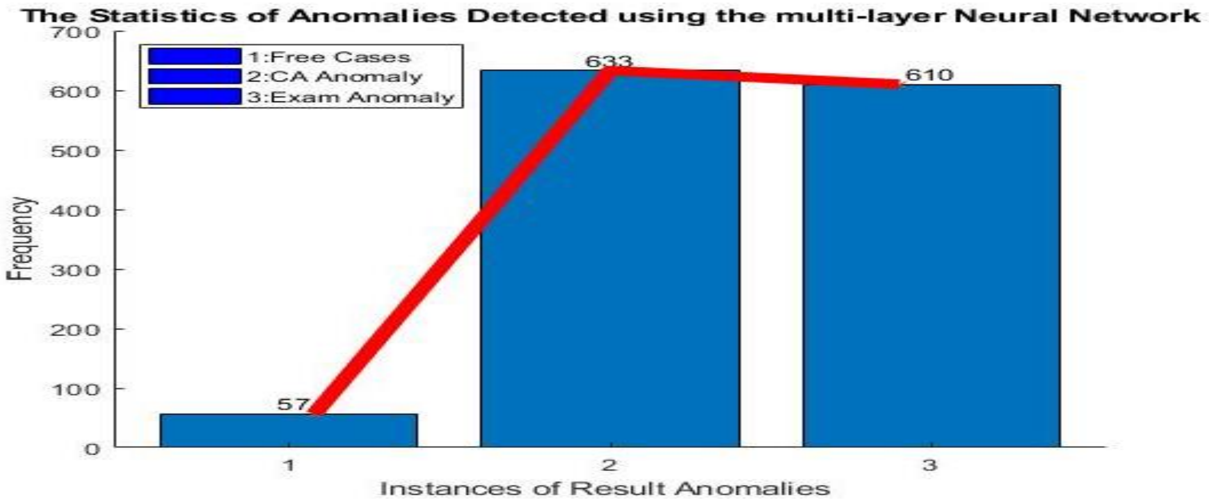


Figure 7: The Statistics of anomaly detected by the proposed model(Neural network model)

In testing the neural network for detection of result anomalies (NNMDRA) system; using one thousand three hundred different dataset contain weighted CA and exam scores generated from MATLAB random function and the results paired suitable with the expected anomalies of the NNMDRA detected CA and exam anomalies as recorded and free cases of irregularities recorded in situations where there are no CA or exam anomalies.

The model accuracy was computed as the ratio of correct classification(True positive and negative) to the overall number of cases using MATLAB function given below as:

$$\text{Accuracy}=\frac{Total\ Number\ of\ Correct\ Classifications}{Total\ Number\ of\ Cases}=\frac{TP+TN}{TP+TN+FP+FN} \quad 9$$

Where TN represent true positive cases = 633, FP is false positive = 76, TP is true positive = 610, FN is false negative cases = 19. The ANN model produced 96% accuracy.

V. Conclusion

The existing method of operation in most organization has proved to be highly ineffective in organizations that deal with result anomaly detection system and maintained fast, effective, reliable and accurate result with the proposed method. Most of the shortcomings of the existing system are corrected by the new system. From the above analysis, we conclude that the proposed system with the aid of neural network feed-forward machine learning algorithm is much better than the existing method (system).

REFERENCES

Aderemi, O. A., and Andronicus, A. A. (2017). A Survey of Machine-Learning and Nature-Inspired Based Credit Card Fraud Detection Techniques. *International Journal of System Assurance Engineering and Management, 8*(2), 937-953.

Adetunmbi, A. O., Samuel O., Falaki, O., Adewale, S., and Boniface K., (2008). “Network Intrusion Detection based on Rough Set and k-Nearest Neighbour”, *International Journal of Computing and ICT Research*, 2(1), 60 - 66.

Alireza O., and Bita S. (2008). Intrusion Detection in Computer Networks Based on Machine Learning Algorithms, *IJCSNS International Journal of Computer Science and Network Security*, 8(11), 19-20

Aneetha, A. S., and Bose, S. (2012). The Combined Approach for Anomaly Detection using Neural Networks and Clustering Techniques, *Computer Science and Engineering: An International Journal (CSEIJ)*, 2(4), 37 – 46.

Andrew, M. S., Pang, W. K., Zhenghao, C., Maneesh, B., Bipin, S. and Andrew, Y. N. (2011). On Random Weights and Unsupervised Feature Learning, *Appearing in Proceedings of the 28th International Conference on Machine Learning(ICML),* Bellevue, WA, USA. 1089–1096.

Balanchi V., Aggarwal, N., and Venkatesan, S. (2015). *Anomaly detection in IPv4 and IPv6 networks using machine learning. 12th IEEE India International Conference (INDICON), New Delhi, India, 1-6*

Fujimaki, R.,Yairi, T., and Machida, K. (2005). An Approach to Apacecraft Anomaly Detection Problem using Kernel Feature Space. Paper Presented at the In Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, NY, USA.

Hamza, O. S.,Ruqayyah, S., and Mohammed, O. (2016). Detecting Anomalies in Students‟ Results Using Decision Trees. *International Journal of Modern Education and Computer Science(MECS)*, 1312–1317. *https://doi.org/10.5815/ijmecs.2016.07.04*

Hawkins, D. M. (2001). The detection of errors in multivariate data using principal components. *Journal of the American Statistical Association, 69*(346), 340–344.

Jamal, H., and Aishwarya, M. (2017). Performance Analysis of Some Neural Network Algorithms using NSL-KDD Dataset, *International Journal of Computer Trends and Technology (IJCTT)*, 50(1), 43-45.

John, E. B.,Derek, T. A., and Chee, S. C. (2017). Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools, and Challenges for the Community. *Journal of Applied Remote Sensing, 11*(4), 609.

Peddabachigari, S.,Abraham, A.,Grosan, C.,(2007). Modeling Intrusion Detection System using Hybrid Intelligent Systems. *Journal of network and computer applications, 30*(1), 114-132.

Ruggieri, S. (2002). "Efficient C4.5 Classification Algorithm", *IEEE Transactions on Knowledge and Data Engineering*, 14(2), 438-444

Ruhui, M. (2008). Network Anomaly Detection Based on Wavelet Fuzzy Neural Network with Modified QPSO, Taylor and Francis Group. 49-60. https://doi.org/10.1080/15501320802540488

Salima O., Asri N., and Hamid H. J. (2013). Machine Learning Techniques for Anomaly Detection: An Overview, *International Journal of Computer Applications ((IJCA)*, 79 (2), 33:41

Singh, S., Kumar, N., and Kaur, N. (2014). Design And development Of Rfid Based Intelligent Security System. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 3*.

Talwar, A., and Kumar, Y. (2013). Machine Learning: An Artificial Intelligence Methodology. *International Journal of Engineering and Computer Science, 2*, 3400-3404.

Vrat, B. N., Aggarwal, S. and Venkatesan, S.(2015). Anomaly DEtection in IPV4 and IPV6 Networks using Machine Learning, 12th IEEE Idian International Conference(INDICON), New Delhi, India, 1-6..