



A Comparative Analysis of Neural Network and Decision Tree Model for Detecting Result Anomalies

Stanley Ziweritin^{1*}, Barilee Barisi Baridam², Ugochi Adaku Okengwu²

¹Department of Estate Management, Akanu Ibiam Federal Polytechnic, Unwana, Nigeria

²Department of Computer Science, University of Port Harcourt, Choba, Nigeria

Email: *stanlo4godsluv@yahoo.com

How to cite this paper: Ziweritin, S., Baridam, B.B. and Okengwu, U.A. (2022) A Comparative Analysis of Neural Network and Decision Tree Model for Detecting Result Anomalies. *Open Access Library Journal*, 9: e8549.

<https://doi.org/10.4236/oalib.1108549>

Received: March 1, 2022

Accepted: March 28, 2022

Published: March 31, 2022

Copyright © 2022 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The decision tree and neural network models are considered as one of the fastest and easy-to-use techniques having the ability to learn from classified data patterns. These models can be employed in detecting result anomalies measurable under normal circumstances on the bases that the student is healthy, had no problem and sat for exams. The existing techniques lack merit and integrity to efficiently detect irregularities found between student continuous assessments and exam scores. The addition of weights and calibrated values aided the learning process and addressed the problem facing the existing methods in operation. This provided an instance of having suitable control over the objective function in overcoming the identified problem. The added calibrated value helped control wrongly classified data patterns and improved the intelligence of the model. In this paper, the K-fold cross-validation test was employed to have a better classification report with the best split. This research was aimed to provide a comparative analysis of neural network and decision tree model for detecting result anomalies. The functionality of both models was used as a measure to check against result anomalies. This resulted in 96% and 91% accuracy with feed-forward multi-layered neural network and decision tree technique.

Subject Areas

Machine Learning

Keywords

Anomaly Detection, Decision Tree, Feed-Forward, Neural Network, Pre-Processing

1. Introduction

Data professionals all over the world are increasingly conducting a systematic search on machine learning models. This helps in providing insights and innovative solutions to real-life challenges or problems. The Neural Network (NN) and Decision Trees (DTs) are all machine learning methods used by data professionals in preference to solve realistic problems. This can also be used to detect irregularities found in student examination results. And this comprises of course work, assignment, practical score, test/quiz score as Continuous Assessment (CA) and exam scores. These depend mainly on the complexity of the model and the problem at hand [1]. Also, NN and DTs use some sets of input data to produce output for decision-making [2]. An NN structure comprises of input nodes, multiple levels of hidden layers and output units that carry out self-optimization through learning with each neuron receiving input and performing an operation. The input accepts data from the external environment process and feeds it to the hidden layer and the hidden layer accepts output from the input. The hidden layer presents it into a form that can be accepted and interpreted by the output layer given to the external environment [3]. The hidden units were introduced as a single layer with feed-forward and back-propagation NN and used random weights to present output as input to the output layers [4]. What are the NNs generally? Unlike traditional gradient-based algorithms which learn network weights slowly and relatively much faster, the NNs also can learn and detect anomalies from classified data patterns to produce the desired result and even though a neuron could not respond as required. The process of training and testing of NN and DT model places a very huge burden like misclassification error and overfitting on the existing methods of operation in detecting CA and exam type of anomalies. The limits that defined normal and abnormal behavior of data are often not precisely defined for one or more data domains. It occurs due to a lack of a well-defined data representation (no standard) which poses challenges for both conventional and some machine learning techniques. The arising need for a large-scale anomaly detection system made it impossible for existing techniques to find outliers (optimal solution) when the volume of training data increases significantly. The aim of this paper is to build a comparative analysis of a NN and DT model for detecting students' result anomalies. This model will be developed based on the condition that the student is healthy, had no problem and sat for exam [5] and the functionality of both models will be used to check against result anomalies. We intend to feed the model with weighted CA and exam scores to compute for the calibrated value, CA and exam type of anomalies after training and testing stages. The addition of weights and calibrated values will help control the objective function in checking against result anomalies.

2. Literature Review

A combination of K-means clustering and decision tree type of Machine Learning (ML) techniques was adopted on some set of training instances to form k clus-

ters in detecting network attacks known as anomalies [6]. This study helped in minimizing False Positive Rate (FPR) and maximized Balance Detection Rates (BDRs) as stated. This required an improvement on false positive rate to control the attacks on users that leads to the use of Naive Bayes and DT for adaptive intrusion detection system. The adoption of the Naive Bayes model with some adjusted gamma (γ) and beta (β) variables made it possible to effectively keep track of the false positives rate for different types of networks attack, and also balanced the detection rates required for user network attacks but produced poor detection rate with the existing dataset [7]. The genetic algorithm (GA) was proposed as self-assessment tool to improve the quality of academic activities after evaluating marks allotted to students. This model was developed and trained with the process of Crossover and Mutation to predict the overall performance of students' at the end of every semester examination. The student assessment model made use of population chromosomes in representing Potential Solutions (PSs) to the identified problem through the fitness function. A future generation was created by selecting more fitted individuals from the population with the use of genetic operators (cross-over and mutation). The cross-over operation created new individuals by combining parts from multiple individuals and mutation operator created new individuals from the old individuals. The fitness function converged and produced accurate results after repeating this process for a number of generations in analyzing subject marks of compiler, automata and data structure). The results of experiment revealed that compiler and data structure was given high importance to reach a better performance but produced poor level of metric accuracy.

A very novel decision tree and support vector machine (SVM) approach was proposed in detecting anomalies [8]. The data set was first allowed to pass through the decision tree and the output of the DT was feed to the support vector machine for finding anomalous content to obtain the desired output. The results revealed to be good in performance with the existing system dataset but when compared to the improved SVM model produced more accurate result. A decision tree result anomaly detection system was developed with classified data instances that connected root, sub-nodes and edges. In their work, the C4.5 type of decision tree model was adopted in the experiment with instances of leaf nodes fixed to 2 and a pruning value of confidence level set to 0.25. The dataset was subdivided into two groups: testing (20%) and training (80%) set, but was unable to detect some cases involving result anomalies at the training stage which resulted in low sensitivity value. Meaning sensitivity value was not accurate and reliable leading to inaccuracy in detection which was affected by the magnitude and variation in dataset. A binary classifier and regression tree model was developed to divide the forest space into certain subsets of tree's with leaf nodes corresponding to different subdivisions, this was determined by a splitting rules in respect to each internal node stated [9]. The step function produced errors and resulted in poor performance with the existing dataset. They adopted the

ID3 type of decision tree in combination with CART and C4.5 trees in predicting the safety of a car when fully loaded with passengers and luggages. The CART produced longer training time complexity with the highest prediction accuracy of about 0.5 seconds and 97.36% than the ID3 and C4.5 type of trees, but required more training time to perform better and resulted to over-fitting problem coursed by missing data. The combination of KNN, SVM and DT models was developed to predict student's performance [10]. The classification report of SVM produced the highest and best accuracy rate of about 95%, followed by DT 93% and KNN recorded to be 92% as the lowest.

The paper is divided into sections as followings: Section One focuses on the introduction, Section Two presents a brief review of some of the previous approaches to the study area and the gap in exploring the proposed model; Section Three, introduces the materials and methods adopted for developing the model; Section Four, focuses on the results and detailed discussion of results; Section Five presents the conclusion to the paper.

3. Materials and Methods

We intend to build on the areas that we have identified some lapses as stated subject to the condition that the student is healthy, had no mental problem and sat for the exams. We are employing the DT and NN model to detect CA and exam types of result anomalies using MATLAB. The addition of weighted CA, exam, and calibrated values will improve the performance of the system in predicting the target (Figure 1).

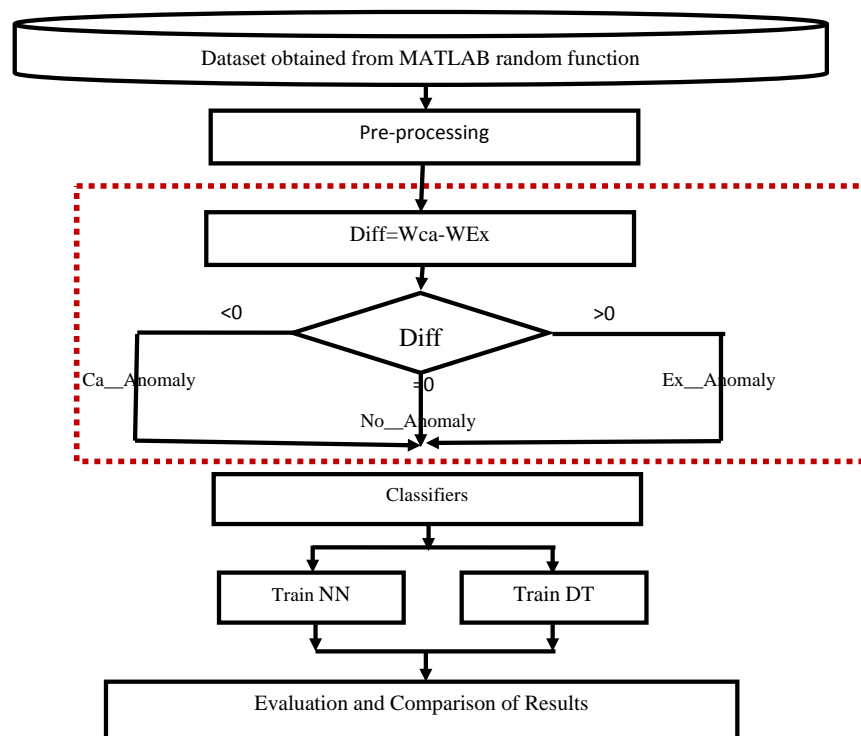


Figure 1. The study strategy.

The architecture is based on a multi-layer feed forward neural network and C4.5 type of DT with students weighted CA and exam scores as input. The weight values lies in between 0 to 1 for controlling the output function set up to work well with a particular set of data type.

3.1. Dataset

The data used by this model is sourced from an experimental dataset of 1,300 items generated randomly using MATLAB random function. This contains instances with detailed information about student's weighed CA and Exam scores as shown below in **Table 1**.

3.2. Pre-Processing

Pre-processing is one of the preliminary stages where data is transformed into computer understandable or acceptable format. Data from the outside world is often inconsistent, incomplete and lacks some behavioral trend and may likely contain errors [11] [12]. In some cases requires scaling data with a standard scale to minimize errors. To build prediction models, we have to convert categorical labels (Ex_Anomaly, CA_Anomaly and No_Anomaly) into numerical labels (1, 2 and 3). But this depends on the method of analysis and both techniques required scaling data to fit the prediction model [13].

3.3. Weighted CA and Exam Scores

The student Continuous Assessment (CA) is set to a maximum of 30% containing test/quiz, assignment and exam set to a maximum of 70% amounted to be 100% and both is weighted to 100% given by the equation below as:

$$\text{Weighted CA (W_CA)} = \frac{\text{CA} \times 100}{30} = \text{CA} \times 3.3 \quad (1)$$

Table 1. Dataset.

S/N	CA	Exam	W_CA	W_Exam	Exact_Diff	TARGET
1	1	22	3	31	-28	CA_Anomaly
2	8	13	27	19	46	Exam_Anomaly
3	17	5	57	8	49	Exam_Anomaly
4	9	43	30	61	-31	CA_Anomaly
5	26	51	87	87	0	No_Anomaly
6	13	43	43	61	-18	CA_Anomaly
7	19	13	63	19	44	Exam_Anomaly
8	30	41	100	59	41	Exam_Anomaly
9	30	52	100	74	26	Exam_Anomaly
10	9	15	30	21	9	Exam_Anomaly
-	-	-	-	-	-	-
1300	9	43	30	61	-31	CA_Anomaly

$$\text{Weighted Exam}(W_Exam) = \frac{CA \times 100}{70} = CA \times \frac{10}{7} \quad (2)$$

The exact difference is the absolute value of the difference between the weighted CA and Exam scores given as:

$$E_Exact = \text{abs}(W_CA - W_Exam) \quad (3)$$

while the decision tree error (DT_{Err}) term is the difference between the exact difference and the decision tree (DT) predicted difference stated as:

$$DT_{Err} = E_Diff - P_Diff_D \quad (4)$$

3.4. Methodology

The NN model is developed using the feed-forward Artificial Neural Network (ANN) and the C4.5 type of DT for detecting CA and Exam anomalies with weighted student's CA (WCA) and exam (W_Exam) scores. The NN and DT models are trained and tested with (80% or 1040) and (20% or 260) of the total to be 1300 items and error rate set to 0.0001. The tolerance error is the ability of a model to learn when the sample data received is corrupted in some way that is common and important. In severally applications it is not possible to access noise-free data. The proposed model is developed in a modular fashion with each module performing a specific task [14] [15].

Algorithm 1: Neural network

Step	Processes involved
1	Initialization of synaptic weights
2	Preprocess input block with CA and exam scores
3	Pass inputs to the real output block (O_{real})
4	Compare if O_{real} matches $E_{expected}$; then successful
5	If O_{real} mismatch $E_{expected}$; then repeat step 3
6	Define boundaries for anomalous data points
7	Change coinciding synaptic weights with target and repeat step 2 until training set is empty
8	Record similar patterns in training set
9	If trained process is complete; then evaluate ANN
10	Return

Algorithm 2: Decision Tree

Step	Processes Involved
1	Start Form_Dcision_Tree (DT)
2	Compute Class Frequency Value and Return Leaf_Node
3	CreateDT of N_nodes with Class Frequency Table (CFT)
4	ComputeGainValue (A) for Each_Attributes of A

```

5      N_Test = Best_Attribute_Gain
6      Test If N_Test Is Continuous Then
7  Find_Threshold_Value
8  If CFT Is Empty Then Child_NNode Is a Leaf_Node
9  If CFT Is Not Empty Then Child_NNode to Form_DT
10 Compute_Error_NNode
11 Return NNode

```

3.5. Neural Network Model

The neural network architecture of the proposed system is made up of two input variables (CA and exam scores), ten hidden layers and one output layer (CA or exam anomalies) as shown in **Figure 2**, with the introduction of a weighted CA (WCA) to control the NN objective function in producing a reliable result. We adopted the use of feed-forward multi-layer neural network as a structure for input and output connections [16] [17]. The NN algorithm is feed with inputs comprises of 80% training and 20% testing data from a total of 1300 items. And we allowed the network to weight and compute the difference between the weighted CA and exam scores with tolerance level set to 0.00001. The idea is to reduce this error rate, until the NN learns from the training data. Finally, we proposed the use of NN activation function to control the output in producing a better and more accurate result.

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^n x_j w_{ji} \quad (5)$$

The output function uses the sigmoid function:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{f(\bar{x}, \bar{w})}} \quad (6)$$

We defined the error function for the output of each neuron as:

$$E_j(\bar{x}, \bar{w}, d) = \sum (O_j(\bar{x}, \bar{w}) - d_j)^2 \quad (7)$$

The weights are all adjusted as follows:

$$\Delta w_{ji} = -\eta \left(\frac{\partial E}{\partial w_{ij}} \right) \quad (8)$$

where x are the inputs, w_{ij} are the weights $O_j(\bar{x}, \bar{w})$ are the actual outputs, d_j are the expected outputs and η is the learning rate.

The pieces in between the input and the output layer are the hidden layers in neural network [18] [19] [20]. The increase in hidden layer does not really affect the accuracy but depends on the complexity of the problem at hand. But this decreases neural network training time. The output of the input is stored and processed by the hidden layer which transforms the output to something that can be used and accepted by the output layer. The output of the hidden layer becomes the input to the output layer as shown in **Figure 2**. The learning process proceeds

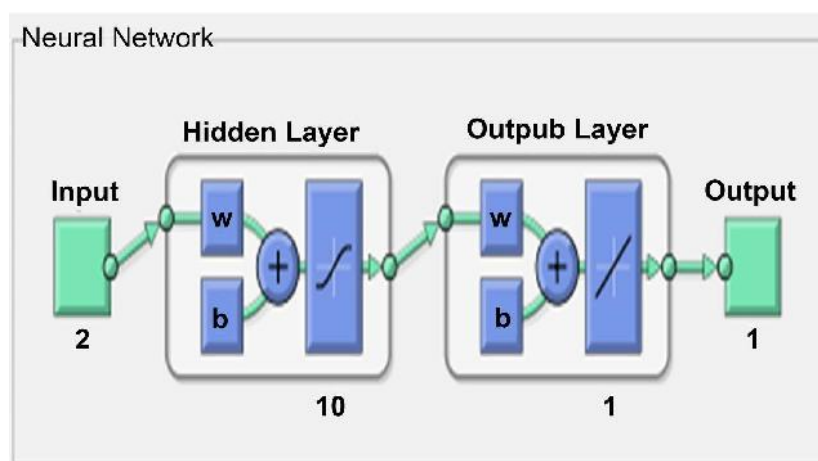


Figure 2. The representation of the NN model.

by ways of presenting the network with a training set consists of input and response patterns. The error values of NN are computed using Equation (3) in comparison with the actual and target output for a given pattern. The error value was then used to alter the connection strength between layers in order to achieve a better network response in subsequent iterations.

3.6. The Decision Tree Model

A decision tree can learn through the concept of divide and conquer by splitting the source dataset into subdivisions called sub-nodes based on the value test [21] [22]. And this process is repeated on each subset in a manner called recursive partitioning. The recursion is completed and terminated when splitting no longer adds value to the predictions. In DT, data arrives in the form of records [23] [24], written as:

$$(x, Y) = (x_1, x_2, x_y, L, x_k, Y) \quad (9)$$

where Y represents the dependent variable taken as the target variable while x is the independent variable as vector containing of the input variables X_1, X_2, X_3, L, X_n , etc. The DT describes a structure for making decisions characterized with an ordered pair of nodes [25] [26]. Each sub-node is associated with a decision function of one or more features generated from IF...THEN rules. There are different types of DTs which includes Classification and Regression Trees (CRT), Chi-Square Automatic Interaction Detection (CHAID) and C4.5 as extension [27] [28]. The training of DT as a binary classifier aid learning with set of instances to classify new instances. The learning rate for each variant is fine-tuned to fall between 0.1 to 1 with specified number of iterations [29].

4. Results and Discussion

We discussed about several machine learning techniques and adopted NN and DT algorithms. The multi-layered feed-forward NN was employed using two input neurons and ten hidden layers to produce a single output of having anomaly

or no_anomaly. The weights, calibrated values and C4.5 type of binary classification tree model was employed to obtain the best split and solution.

Figure 3 depicts the total number of dataset tested with their respective classes of result anomaly and frequency. The model automatically group results into their anomaly classes as Ex_Anomaly, CA_Anomaly and No_Anomaly (Free cases) shown in **Figure 1**. The Ex_Anomaly recorded about 660, CA_Anomaly = 629 and free cases of anomaly = 11. The summation all the classes produced the total dataset, *i.e.* $660 + 629 + 11 = 1300$ items.

Figure 4 depicts the decreasing Mean_Square_Error (MSE) validation graph for training and testing stages and terminates at its tolerance level within the iterations. There was no sign of over-fitting or under-fitting because the MSE of

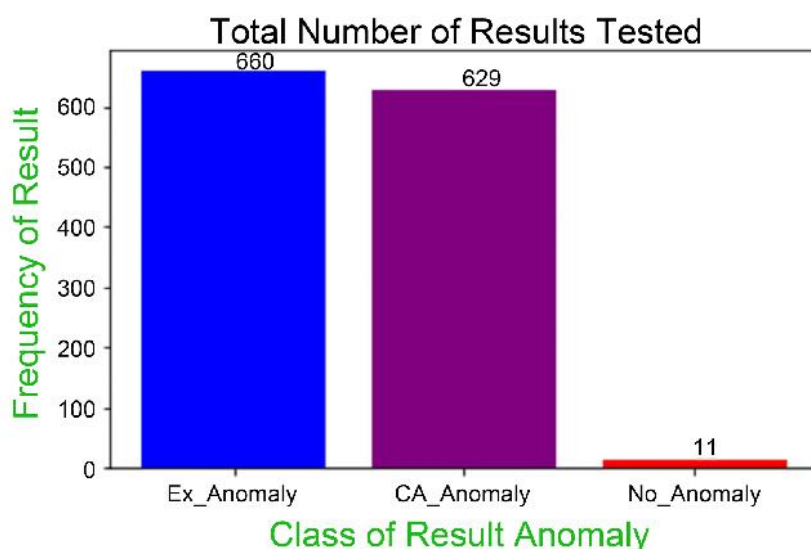


Figure 3. Total number of tested results.

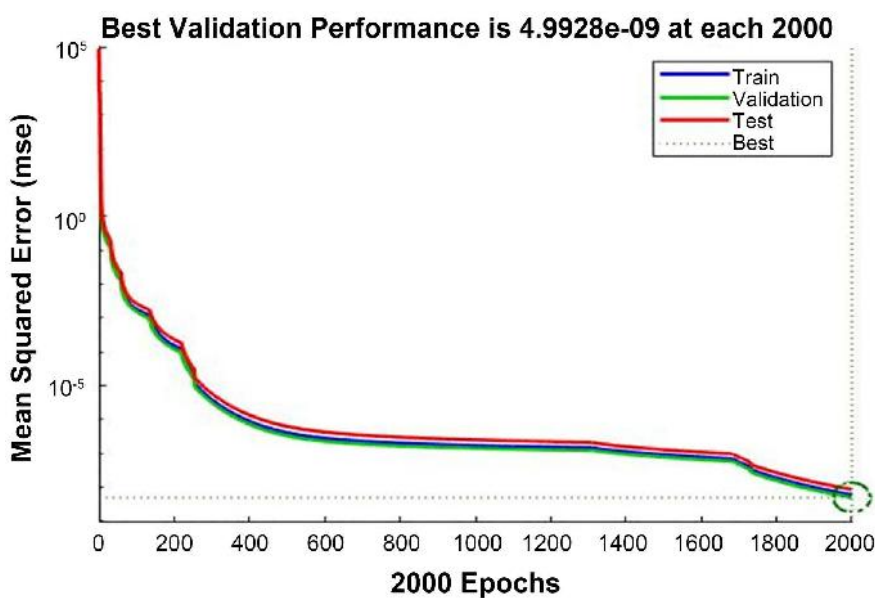


Figure 4. The performance plot of NN.

validation decreases and coincide with the best possible solution at 2000 epoch with colored circle. The NN was able to learn with high accuracy and best validation performance value of $4.9928\text{e}-09$ in predicting the target.

Figure 5 shows the graph of DT estimated objective function. The estimated objective function ranges from 1, 0.95, 0.9, to 0.6 against the minimum leaf node size ranges from 10^0 , 10^1 , 10^2 to 10^3 which decreases and convergences within the work space. The minimum feasible point occurs exactly at 0.65 in the Y axis, set of observed points, model error mean occurs within 10^0 to 10^2 of the horizontal axis.

The function evaluation graph shows the convergences of the minimum observed objective function. The objective function increased and decreased within the range of 0 to 5, but constant and linear from 5 to 30 at the horizontal axis to cut the Y-axis at 0.65 and was able to detect CA and exam anomalies within the work space showing the distance for the whole spectrum of iterations for the dataset (**Figure 6**).

Figure 7 shows the tree structure of the proposed DT gotten from the classified dataset feed into the model with the concept of recursive partitioning. This generated a root, sub and leaf nodes in a tree like fashion with some predefined rules using MATLAB. The decision tree shown in **Figure 7** is constructed and grown by the following IF ... THEN statement using the DT predicted difference (DT_Preded_Diff). This clearly defined the anomaly class boundaries as added advantage over the existing methods.

```

If (DT_Predicted_Diff > 0) THEN Exam_anomaly
ELSEIF (DT_Predicted_Diff < 0) THEN CA_anomaly
ELSE:
    Normal_Case or Free_cases
End IF

```

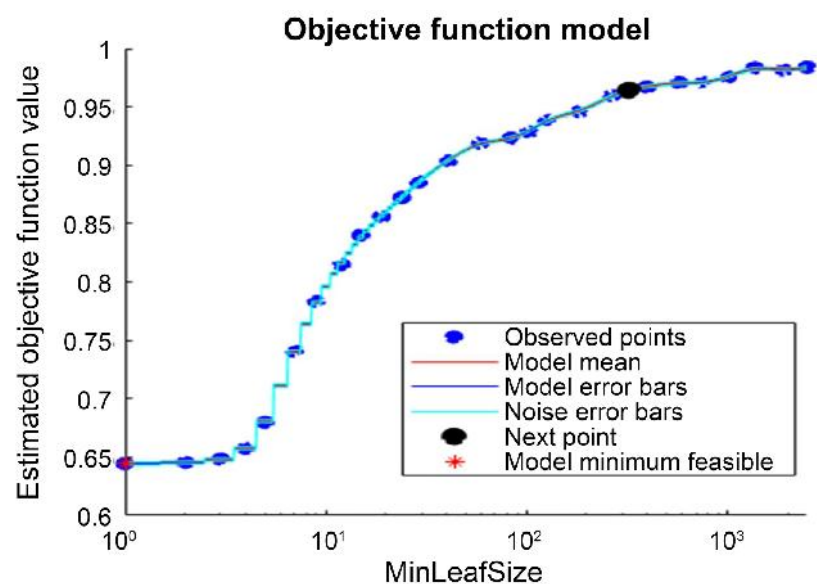


Figure 5. The DT convergence graph of the objective function.

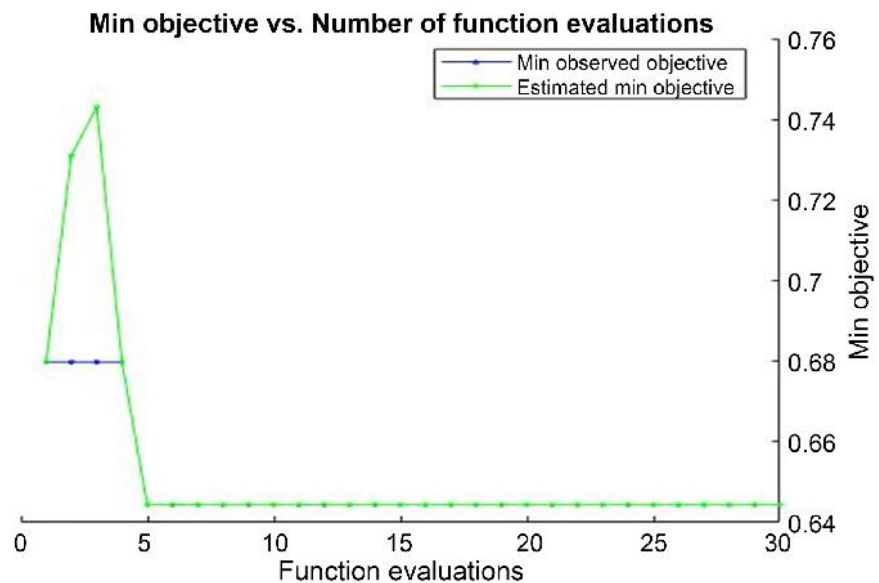


Figure 6. The DT function evaluation graph of the objective function.

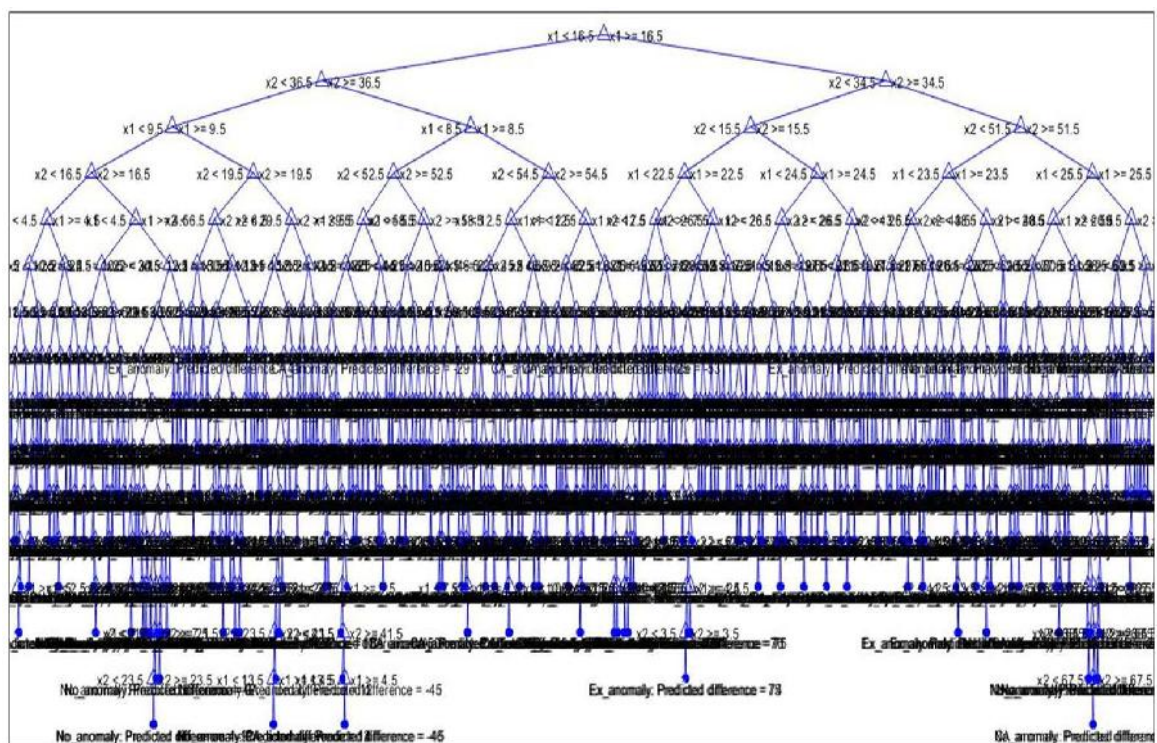


Figure 7. The decision tree of the proposed system model.

Comparing the Results of the NN and DT Model

A comparative study of the proposed system was carried out with a total of dataset of 1300 items containing students *CA*, *exam*, weighted *CA* and *exam* scores, exact difference, NN and DT predicted differences, DT error term and anomaly detected by both models given bellow in **Table 2** for the first data items.

Table 2. Comparing the proposed and existing system model.

S/N	CA	Exam	W_CA	W_Exam	E_Diff	ABS (E_Diff)	Neural Network (NN)		Decision Tree (DT)		
							P_DiffN	ANOMALY	P_DiffD	Err	ANOMALY
1	1	22	3	31	-28	28	28.429	CA_Anomaly	25	3	CA_Anomaly
2	8	13	27	19	8	8	8.4285	Exam_Anomaly	8	0	No_Anomaly
3	17	5	57	8	49	49	48.857	Exam_Anomaly	51	2	Exam_Anomaly
4	9	43	30	61	-31	31	31.429	CA_Anomaly	35	3	CA_Anomaly
5	26	51	87	87	0	0	0	No_Anomaly	3	3	CA_Anomaly
6	13	43	43	61	18	18	18.429	CA_Anomaly	19	1	No_Anomaly
7	19	13	63	19	44	44	44.429	Exam_Anomaly	44	0	Exam_Anomaly
8	30	41	100	59	41	41	41.429	Exam_Anomaly	38	3	Exam_Anomaly
9	30	52	100	74	26	26	25.714	Exam_Anomaly	21	4	Exam_Anomaly
10	9	15	30	21	9	9	8.5714	Exam_Anomaly	7	2	Exam_Anomaly
-	-	-	-	-	-	-	-	-	-	-	-
1300	9	43	30	61	-31	31	31.429	CA_Anomaly	35	3	CA_Anomaly

Table 2 shows how the proposed model was able to detect CA and Exam anomalies. The weighted students' CA (W_CA) and exam (W_Exam) values are all computed by the NN and DT algorithms. The exact difference (E_Diff) or solution = abs (W_CA-W_Exam) while the DT error (Err) value= (E_Diff – P_DiffD) shown in **Table 2** obtained from the first ten data items. By comparing what the Neural Network (NN) predicted as the difference (P_DiffN) = (28.429, 8.4285, 48.857, 31.429, 33, 18.429, ...) and the DT predicted difference (P_Diff) = (325, 8, 51, 35, 36, 19, 44, 38, ...) with the attributes of anomaly. This revealed that the NN produced a better, precise and more reliable result than the Decision Tree (DT) in terms of detected anomalies.

The following were detected by the NN and DT model shown in the overlapping chart (**Figure 8**) using MATLAB. The free case of result anomalies registered by DT is 116 which grow higher than the NN recorded as 97 free cases. The instances of exam anomalies detected by DT are 605 totally overlapped by number detected by the NN which is 633 cases. Also the total number of CA anomalies detected by DT shown as 579 and totally overlapped by the NN as 610 cases of CA anomalies. Total number of training dataset = 1300.

The accuracy of the proposed model is the ratio of correct classifications (True positives and negatives) from the overall number of cases using MATLAB function given below in the equation as:

$$\text{Accuracy} = \frac{\text{total number of correct classification}}{\text{total number of cases}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (10)$$

where TN represents true negative = 579, FP is false positive = 97, TP is True positive = 605 FN is false negative = 19 cases and produced 91% level of accuracy for the DT model. While TN = 633, FP = 37, TP = 610 FN = 20 cases and produced 96% level of accuracy for the NN model.

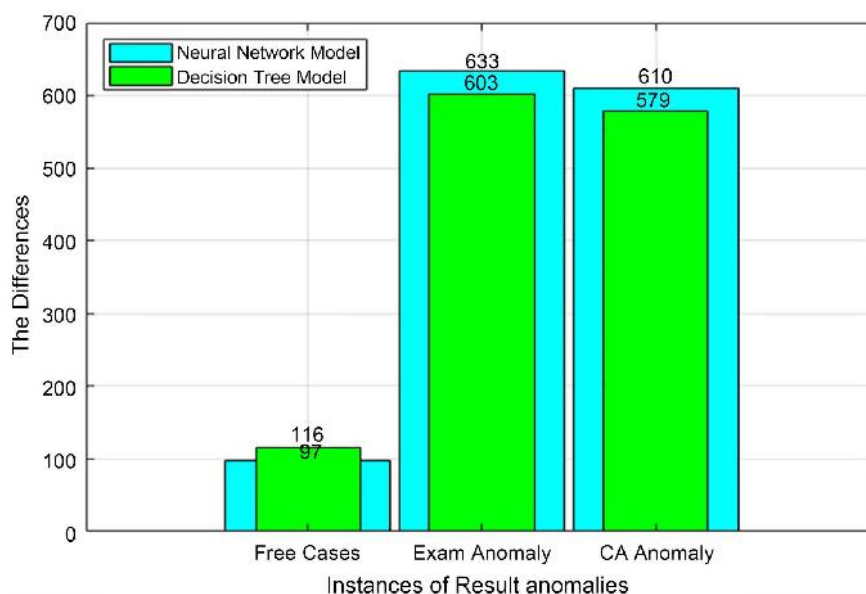


Figure 8. The overlapping statistics of NN and DT model.

5. Conclusion

The error value for both systems converged within the selected types of result anomalies. From the experimental results, DT recorded a wider rate of error compared to NN model. Also, the error rate of NN converged faster and produced a higher accuracy rate than DT when used to check against result anomalies. The added weights and calibrated values reduced the error value and improved the general performance of the proposed system. Therefore, we evidently conclude that the NN model performed better in terms of accuracy and precision in detecting anomalies.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Hawkins, D.M. (2001) The Detection of Errors in Multivariate Data Using Principal Components. *Journal of the American Statistical Association*, **69**, 340-344. <https://doi.org/10.1080/01621459.1974.10482950>
- [2] Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2016) Extreme Learning Machine: Theory and Applications. *International Journal of Neurocomputing*, **70**, 489-501. <https://doi.org/10.1016/j.neucom.2005.12.126>
- [3] 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**, 937-953. <https://doi.org/10.1007/s13198-016-0551-y>
- [4] Talwar, A. and Kumar, Y. (2013) Machine Learning: An Artificial Intelligence Methodology. *International Journal of Engineering and Computer Science*, **2**, 3400-3404.
- [5] 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)*, **8**, 1312-1317. <https://doi.org/10.5815/ijmecs.2016.07.04>
- [6] Farid, D.M., Harbi, N. and Rahman, M.Z. (2010) Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection. *International Journal of Network Security and Its Applications*, **2**, 12-25. <https://doi.org/10.5121/ijnsa.2010.2202>
 - [7] Peddabachigari, S., Abraham, A. and Grosan, C. (2007) Modeling Intrusion Detection System Using Hybrid Intelligent Systems. *Journal of Network and Computer Applications*, **30**, 114-132. <https://doi.org/10.1016/j.jnca.2005.06.003>
 - [8] Jha, J. and Ragha, L. (2013) Intrusion Detection System Using Support Vector Machine. *International Journal of Applied Information System (IJ AIS)*, **5**, 25-30.
 - [9] Joseph, J.F., Das, A. and Seet, B.C. (2011) Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA. *IEEE Transaction on Dependable and Secure Computing*, **8**, 12-23. <https://doi.org/10.1109/TDSC.2009.48>
 - [10] Wiyono, S. and Abidin, T. (2019) Comparative Study of Machine Learning KNN, SVM, and Decision Tree Algorithm to Predict Student's Performance. *International Journal of Research*, **7**, 1-13. <https://doi.org/10.29121/granthaalayah.v7.i1.2019.1048>
 - [11] Abdellatif, H., Mohamed, S. and Mohamed, F. (2016) Face Recognition: Synthesis of Classification Methods. *International Journal of Computer Science and Information Security (IJCSIS)*, **14**, 5-11.
 - [12] Abhishek, K. and Anil, P. (2019) Implement of Students Result by Using Genetic Algorithm. *International Journal of Computer Sciences and Engineering*, **7**, 51-56.
 - [13] Anyanwu, M.N. and Shiva, S.G. (2009) Comparative Analysis of Serial Decision Tree Classification Algorithms. *International Journal of Computer Science and Security (IJCSS)*, **3**, 37-46.
 - [14] 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**, 37-46. <https://doi.org/10.5121/cseij.2012.2404>
 - [15] Badr, H.H.E., Abdelkarim, M. and Mohammed, E. (2014) A Comparative Study of Decision Tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications*, **4**, 13-19. <https://doi.org/10.14569/SpecialIssue.2014.040203>
 - [16] Zhou, Z.H., Jiang, Y., Yang, Y.B. and Chen, S.F. (2002) Lung Cancer Cell Identification Based on Artificial Neural Network Ensembles. *Artificial Intelligence in Medicine*, **24**, 25-23. [https://doi.org/10.1016/S0933-3657\(01\)00094-X](https://doi.org/10.1016/S0933-3657(01)00094-X)
 - [17] Ziweritin, S., Baridam, B.B. and Okengwu, U.A. (2020) Neural Network Model for Detection of Result Anomalies in Higher Education. *Scientia Africana: An International Journal of Pure and Applied Sciences (IJPAS)*, **19**, 91-104.
 - [18] 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**, 609. <https://doi.org/10.1117/1.JRS.11.042609>
 - [19] Omlin, C.W. and Snyders, S. (2003) Inductive Bias Strength in Knowledge-Based Neural Networks: Application to Magnetic Resonance Spectroscopy of Breast Tissues. *Artificial Intelligence in Medicine*, **28**, 30-54. [https://doi.org/10.1016/S0933-3657\(03\)00062-9](https://doi.org/10.1016/S0933-3657(03)00062-9)
 - [20] Yedjour, D., Yedjour, H. and Benyettou, A. (2011) Explaining Results of Artificial Neural Networks. *Journal of Applied Sciences*, **2**, 1-45. <https://doi.org/10.3923/jas.2011.2855.2860>
 - [21] Balogun, A.O., Mabayoje, M.A., Salihu, S. and Arinze, S.A. (2015) Enhanced Classification via Clustering Techniques Using Decision Tree for Feature Selection. *International*

Journal of applied Information System (IJAIS), **9**, 11-16.

<https://doi.org/10.5120/ijais2015451425>

- [22] Bhargava, N., Sharma, G., Bhargava, R. and Mathuria, M. (2013) Decision Tree Analysis on j48 Algorithm for Data Mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, **2**, 45-98.
- [23] Patel, H.P. and Prajapati, P. (2018) Study and Analysis Tree Based Classification Algorithms. *International Journal of Computer Sciences and Engineering (IJCSE)*, **6**, 74-78.
<https://doi.org/10.26438/ijcse/v6i10.7478>
- [24] Soranamageswari, M. and Meena, C. (2020) Histogram Based Image Spam Detection Using Back Propagation Neural Networks. *Global Journal of Computer Science and Technology (GJCST)*, **9**, 62-67.
- [25] Lakshmi, T.M., Martin, A., Begum, R.M. and Venkatesan, V.P. (2013) An Analysis on Performance of Decision Tree Algorithms Using Students Qualitative Data. *International Journal of Modern Education and Computer Science (IJMECS)*, **5**, 18-27.
<https://doi.org/10.5815/ijmeecs.2013.05.03>
- [26] Li, Y., Xing, H., Hua, Q. and Wang, X. (2014) Classification of BGP Anomalies Using Decision Trees and Fuzzy Rough Sets. *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, San Diego, 5-8 October 2014, 1312-1317.
<https://doi.org/10.1109/SMC.2014.6974096>
- [27] Chien-Liang, L. and Ching-Lung, F. (2019) Evaluation of CART, CHAID, and QUEST Algorithms: A Case Study of Construction Defects in Taiwan. *Journal of Asian Architecture and Building Engineering*, **18**, 1-40.
<https://doi.org/10.1080/13467581.2019.1696203>
- [28] Zhang, X. and Jiang, S.A. (2012) A Splitting Criteria Based on Similarity in Decision Tree Learning. *JSW*, **7**, 82-1775. <https://doi.org/10.4304/jsw.7.8.1775-1782>
- [29] Huang, M. and Hsu, Y. (2012) Fetal Distress Prediction Using Discriminant Analysis, Decision Tree, and Artificial Neural Network. *Journal of Biomedical Science and Engineering (JBSE)*, **5**, 526-533. <https://doi.org/10.4236/jbise.2012.59065>