International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 11, Issue 6, November –December 2022 ISSN 2278-6856

Random Forest Based Classifiers for Detecting Result Anomalies

[1]Stanley Ziweritin, [2]Iduma Aka Ibiam

[1,2]Department of Estate Management and Valuation,
Akanu Ibiam Federal Polytechnic, Unwana, Ebonyi State, Nigeria

Abstract*: Random forest(RF) is a supervised machine learning approach that experts use to build and integrate many decision trees into a single forest. It takes considerable expertise to detect result anomalies depending on the degree of disparity between students' CA and exam scores. It is doable to train RF-based classifiers to accurately identify anomalies with imbalanced data categorization. The aim is to develop RF-based classifiers capable of detecting abnormalities in student results, such as when a student performed remarkably well on the exam but poorly on the CA, or vice versa. The SMOTE technique was used to resolve unbalanced data categorization, which helped reduce dataset bias toward the majority class while also ensuring that the minority class received an acceptable sample size. Strong decision-makers were grouped into a class of majority vote using the grid search and randomized function. Trees' capacity to learn from small data samples was arbitrarily constrained by the uniformly distribution function, which increased model accuracy and reduced tree correlation. Comparatively, the Classification, Adaboost, and GradientBoosting classifiers produced accuracy scores of 99.00%, 95.17%, and 81.50% respectively.*

Keywords: Anomaly detection, Bagging, gradient boosting, Random Forest, classifier

1. INTRODUCTION

Random forest is one of the most popular known and successful supervised machine learning(ML) techniques used by professionals to carry out prediction and classification tasks**[1].** The method generates a forest with a predetermined number of decision trees**[2]**,**[3]**. Generally speaking, the more trees we construct and grow, the more reliable the classifications and predictions made. According to C**haudhari and Patil**[4]; It is possible to claim that each tree in the forest "votes" for a particular class by providing a classification to group new objects based on qualities. The forest chooses a classification having the most votes of all other trees in the forest**[5]**. In the case of regression according to **Kulkami**[6]**.** It computes the mean of all outputs from various trees. The RF is a forest that can be developed using a bootstrapped sample and only taking into account a portion of the variables at each phase**[7]**. According to Aggarwal**[8]**, RF can outperform other ML methods because of its diversity. The good news is that RF greatly improves accuracy by combining flexibility with decision trees' simplicity**[9]**. The numerous academic board meetings that are held to approve results take up a lot of time, and occasionally departments, school boards, and senates find them to be difficult. This is because some academic staff do not follow the established criteria with the capabilities to complete such a task under normal circumstances, it takes considerable expertise to take action depending on the degree of disparity between students' CA and exam scores. Despite the widespread use of ML algorithms in everyday life, this paradigm has had little success when applied to the research field. The existing systems' accuracy metrics were harmed by the usage of non-standard anomalous data classes and the minoring of strong DT-learners in the majority class**[10]**. Automatically searching for strong leaners to be in the majority class is possible with RF-based classifiers. This can be achieved by choosing capable learners during the production of DTs using tweaking hyper-parameter settings.

The aim is to build an efficient result anomaly model in detecting result anomalies using RF-based classifiers. The RF ensemble method provides a better accuracy rate with low error value, highly consistent, reduce bias and variance errors. The grid search technique is be used together with RF, and randomization to improve performance. We are proposing to solve the problem of imbalanced dataset classification using SMOTE technique. The first and second levels of randomization will be added to arbitrary limit trees' ability to learn from the few available samples at each split. This will help prevent DT correlation and lead to improved accuracy. This was accomplished by testing the effectiveness of the Adaboost, RF classification, and gradient Boost models against detected result anomalies. To further increase the detection accuracy, we intend to incorporate equally weighted scores into the RF method and apply weight values to various trees. This will improve model performance and increase detection accuracy. This will be achieved by evaluating the model classifiers' performance against detected result abnormalities.

This paper is organized into the following sections: Section 1 offered an introduction; the section 2 provides a brief assessment of prior approaches related to the subject topic and the gap in studying the proposed model, while the model's materials and methods are introduced in Section 3; the results and a thorough discussion of the results are covered in Section 4; and the paper's conclusion is given in Section 5.

2. LITERATURE REVIEW

Prashanth *et al.*[11], combined random forest classification algorithm with some feature selection techniques to find anomalies (attacks) in computer networks. As more trees are added to the RF, the model's

rate of false positives decreases**[12]**. This was observed during training with varied numbers of trees in the forest space. Numerous different random forest algorithms exist, including GradientBoost, AdaBoost, Bagging, and others. **Primartha & Tama**[13] created multilayer perceptron(MLP) neural networks and RF-based classifiers with 10-fold cross validation test to find anomalies in IoT networks. The k-th of the 10 epochs made use of the training set, which had 175,341 instances with 42 attributes. But this was not applicable in detecting result anomalies. However, this might not be useful in spotting anomalies in student results. To check for abnormal data content, **Brueing** *et al.*,[14] used RF-based anomaly detection methods. The datasets utilized to train and test the model came from the Kaggle repository website and included information on breast cancer, heart disease, and the corona virus, 50%, 87% and 75%, respectively, were recorded for the metrics of RF classification model. ***Salami****, et al.[15]*; suggested using a dataset derived from duly approved departmental student outcomes to discover anomalies using a j48 and C4.5 kind of decision tree. Instances for the number of minimum leaf nodes and pruning after training hyper-parameter values were set to 0.25 and 2 respectively. The model had a good accuracy rate for detecting some of the chosen categories of result abnormalities. However, the dataset obtained was insignificant (reported as having low data quality) and led to an issue with unbalanced data classification, which had an impact on the functionality of the system. Additionally, it was difficult to distinguish between the boundaries of CA and exam anomalies.

2.1 Anomaly detection

Anomaly detection, according to **Callegari** *et al.*[16], is the identification of unusual things, occurrences, or observations that significantly deviate from the majority of the data and hence do not fit a predetermined definition of normal behavior. Applications for anomaly or outlier detection can be found in a wide range of fields, including cyber-security, health, computer vision, statistics, and neuroscience etc**[17]**. To help with descriptive statistics, such as when calculating the mean or statistical variations, anomalies were originally looked for in the data which have been clearly rejected or omitted. Anomalies were eliminated to improve predictions from models like linear regression, and more subsequently, their elimination improved the effectiveness of ML techniques**[18]**. Anomaly detection methods can be divided into three categories: supervised, semi-supervised, and unsupervised ML methods.

(a). Supervised anomaly detection: The supervised anomaly detection approach requires datasets that are labeled as normal and anomalous and involves training a classifier due to the general scarcity of labeled data availability and the intrinsically imbalanced nature of the data classes**[19]**. **Roplekar & Buradkar**[20] created a security solution that uses machine learning(ML) techniques to detect anomalies in a huge computer networks and lessen the issue of false positive rates. The model was preprocessed using attribute selection technique after being trained using data from firewall logs. Based on the established patterns, outliers were identified in the input data.

(b). Semi-supervised anomaly detection: The semi-supervised anomaly detection techniques presumptively use some labelled(tagged) data**[21]**. **Bhadri** *et al.*,[22] compared the operation of the local outlier factor (LOF) and auto-encoder (AE) semi-supervised anomaly detection methods to search for abnormal data points. The auto-encoder recorded a success rate that was much below average based on the accuracy metrics of LOF(41% and 50%). It was unreliable and in need of improvement because the overall performance was below average.

(c). Unsupervised anomaly detection: Unsupervised anomaly detection approaches are among those that are extensively employed because of its broader and more relevant application**[23]**. It starts with the premise that the collection is unlabeled, which are normal, and search for points that deviate from the normal data points. **Ziweritin** *et al.*,[24**]** used the neural network(NN) technique with two inputs, four hidden layers, and one output layer in comparison to the exact value for finding anomalous results. The model was trained, tested and validated to ascertain 91% accuracy level with the exact solution. The model was able to detect CA and exam anomalies in student results. But was unable to deliver metrics with high percentages and demanded extra training time.

2.2 Result anomalies

Anomaly is defined by **Felipe** *et al.*,[25] as any data point that deviates from what is considered normal, expected, and aberrant. Abnormalities found in student test results needs to have a more thorough explanation, but it's vital to remember that anomalies aren't always bad. For instance, the performance of the students in the course cannot be characterized as poor when 36 out of 40 students receive an A. However, the circumstance is viewed as an abnormality because that course has an exceptionally high percentage of A's**[26]**. The forms of anomalies related to course-based and student-based results were examined**[27]**.

(a). Course-based result anomalies: These anomalies concern the overall performance of all students in a certain course**[28]**. The following three course-based anomalies were found: When a large fraction of the class fails a course, it is said to have a high failure rate anomaly. Low grades result anomalies that occur when student's consistently perform poorly in a course. It should be noted that students can have low grades in a course without necessarily failing it, and that there can also be an anomaly known as too many good grades when grater percentage students score high grades in a subject**[29]**.

(b). Student-based result anomalies: These discrepancies are linked to some students' performance in a certain course**[30]**. Students' continuous assessment (CA) score, which ranges from 0-to-30% or 40%; their exam score, which ranges from 0-to-70% or 60% as the case may be; and their total score, which is the sum of their CA and exam scores, can all be used to describe their success rate in a specific course. Five student based abnormalities have

been identified, namely: disproportionate CA versus examination scores anomaly denotes a discrepancy between student's CA and exam scores, as in cases where a student fared exceptionally well on the exam but poorly on the CA, or vice versa. When a student scores 40 out of 40 marks in the CA, it is considered to be an anomalous perfect score**[31]**. When a student earns 70 out of a possible 70 on the exam, this oddity known as the perfect exam score occurs. The borderline failure anomaly occurs when a student fails a course with a final grade of 39, which is one below the passing level of 40.

3. METHODOLOGY

We are combining various RF-based classifiers, including bagging, gradient boosting, and classification models. This work demonstrates the efficiency of RF-based classifiers in result anomaly detection by estimating the class probabilities and combining several weak decision tree learners to become strong learners. The following stages make up the implementation phase: dataset, pre-processing, weighting of CA and exam scores, model development, and evaluations.

3.1 Dataset: The proposed system dataset was gathered from the departments of Akanu Ibiam Federal Polytechnic, Unwana (A.I.F.P.U), Afikpo Ebonyi State, Nigeria, and it included student scores from Continues assessment(CA) and exam(EX). The student CA comprises of assignments, course work, practical scores and class work amounting to 40% and exam score(60%).

Pre-processing stage: is one of the early steps where data is transformed into a format that a computer can interpret**[32]**. Data gathered from the outside world is frequently incoherent, lacking in behavioral trends, and May even contain errors. This requires scaling data with a standard scaler to minimize errors. We must weigh student CA and exam scores equally in order to develop and train the model, and then use the difference to identify the anomalous border.

(a). Weighted CA and Exam scores: Equations 1 and 2 below are used to standardize the anomalous boundary. The student CA is set to a maximum of 40%, with the test/quiz, practical, assignment, and exam set to a maximum of 60%, adding up to 100%. The zero weights represent free case anomalies that result from rule 1, the positive weights represent exam abnormalities, and the negative weights represent CA anomalies as shown in table 7 under the column called "DIFF".

$$\text{Weighted CA(WCA)} = \frac{CA \times 100}{40} = \text{CA} \times \frac{10}{4} \quad 1$$

$$\text{Weighted Exam(WEX)} = \frac{CA \times 100}{60} = \text{CA} \times \frac{10}{6} \quad 2$$

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

$$\text{Diff} = (\text{WCA} - \text{WExam}) \quad 3$$

The decision rule 1 is used to produce the target value:

Rule 1: To have positive and negative weighted values

Step	Line statement
1	If Diff >0:
2	EX_anomaly
3	elif Diff<0:
4	CA_anomaly
5	else:
6	Free_cases
7	Endif

Training and testing data: The total dataset was split into a training set and a testing set using the Spyder IDE and the Python ANACONDA sklearn function. This is a modular program that returns many values (x_test, x_train, y_test, and y_train), with each module handling a different task.

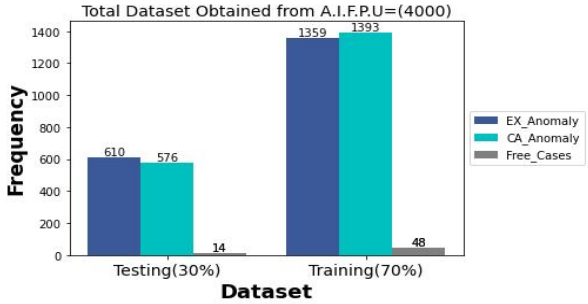

Figure 1: System Dataset

The training set (70% or 2800) and testing dataset (30% or 1200) of the total 4000 items. The testing dataset has a class count of 400 items, made up of 610 exam anomalies, 576 CA anomalies, and 14 free cases (classes without anomaly). The training dataset includes 48 free instances, 1393 CA Anomalies, and 1359 Ex Anomalies colored blue for a total of 2800 items, as is seen in figure 1. The classification of data between CA/Exam anomalies and free cases is highly unbalanced, which could have an impact on model performance.

Random forest: RF is an ensemble technique that assembles a set of extremely randomized classification or regression trees from a set of randomly chosen training data samples and trees are chosen in the process of production**[33]**. If the training dataset contains "N" cases but with substitute; and then each tree is built by randomly selecting 'N' cases from the underlying data. When there are M-input variables, the variables are chosen so that m<M is satisfied at every node. 'M' variables are drawn at random from the 'M' sample, and the node is split using the optimal split on these m. At the forest's producing trees, the values of 'm' are kept constant. RF models are extremely potent, because they have the ability to overcome over-fitting without significantly increasing bias-related error**[34]**. The utilization of several data samples during training could help RFs reduce variation**[35]**.

Gradient Boost classifier: is a well-known technique that experts in ML frequently employ since it is ten times faster than other gradient techniques. Ranking is done using a bespoke loss function which can be used to strength weak DT learners. This depends on the loss function, which should be differentiable. To bring weak DT learners into production, we are balancing computation time with generalization of model performance on testing set. Optimizing hyper-parameter values requires active decisions such as learning rate, maximum number of leaf

nodes and number of estimators. We adopted the gradient boosting with grid Search technique to search for the best hyper-parameter values, strengthen weak learners(decision trees It will assist in correcting any learning rate problems and regulate model behavior. To reduce the overall error, we intend to fit residual samples with small learning rates, which call for more estimators**[36]**.

AdaBoost(Adaptive Boosting): is a suggested ensemble enhancing classifier for combining different classifiers to improve classification accuracy. The AdaBoost is trained using different weighted training samples of CA and exam scores, and it strives to minimize training error in each epoch to give a high level of sample fit. Incorrectly classified observations are given higher weighted CA and test scores in order to increase their likelihood of classification in the following iteration**[37]**. This approach is repeated until all training data is correctly fitted.

Randomized grid search: The randomized search technique yields the best results when combining hyper-parameters at random. This enhanced model performance and guaranteed model accuracy by utilizing every possible combination conceivable**[38]**. The following code snippet was used to implement the randomized search:

```
664
665  from sklearn.ensemble import RandomForestClassifier
666  from sklearn.model_selection import RandomizedSearchCV
667  param_distributions = {
668      "n_estimators": [1, 2, 3, 5, 10, 15, 20, 50, 80,
669                       100,150, 200, 300, 400, 500],
670      "max_leaf_nodes": [2, 5, 10, 20, 50, 100]
671  }
672  search_cv = RandomizedSearchCV(
673      RandomForestClassifier(),
674      param_distributions=param_distributions,
675      scoring="neg_mean_absolute_error",
676      cv=3, n_iter=14, random_state=32,
677      n_jobs=-1, return_train_score=True
678  )
679  search_cv.fit(X_train, y_train)
680
```

SMOTE technique: One of the most well-known oversampling and under-sampling algorithms is SMOTE, which creates its dataset using the concept of nearest neighbors. SMOTE technique utilizing k nearest neighbor is required to achieve a balance between the majority and minority classes. The suggested method is utilized to address the unbalanced class sampling brought on by over- and under-sampling. The proposed approach is only used to the training dataset in order to allow our algorithm to be correctly fitted to the data; the test dataset is left unaltered to accurately reflect the real data. We defined instances with default values to fit, balance, and apply in a single step to provide a changed version of the data. This described the new altered class distribution, which was expected to be balanced by the development of numerous new synthetic data in the minority class. The suggested dataset was resampled using the SMOTE approach, which was implemented in the Python code snippet below:

```
74
75  import pandas as pd
76  from imblearn.over_sampling import SMOTE
77  from sklearn.ensemble import RandomForestClassifier
78  from sklearn.model_selection import train_test_split
79
80  sm = SMOTE(random_state=42, k_neighbors=5)
81  X_res, y_res = sm.fit_resample(X, y)
82  X_train, X_test, y_train, y_test = train_test_split(X_res, y_res,
83                                                      test_size=0.3,
84                                                      random_state=42)
85  rf = RandomForestClassifier(n_estimators= 5, random_state=0)
86  rf.fit(X_train, y_train)
87  y_pred = rf.predict(X_test)
88
89
```

The log function: The log uniformly(Loguniform) distributed function was used to evenly sample data between log(a = learning_rate$_1$) and log(b = learning_rate$_2$), where (learning_rate$_1$ = 10.01) and (learning_rate$_2$ =1), in order to assume even probability for all values with a particular range of model. It enhances grid search methodology and is beneficial for investigating values that fluctuate across different range of magnitude. This allowed us to quickly scan a wide range of tweaking hyper-parameter values and reduce the range to determine the ones that perform best for our model and data.

For this class, the probability density function is:

$$F(x,\text{learning\_rate1},\text{learning\_rate2}) = \frac{1}{x\log\left(\frac{learning\_rate_1}{learning\_rate_2}\right)} \quad (5)$$

For learning_rate$_1$ ≤ x ≤ learning_rate$_1$, learning_rate$_2$ > learning_rate$_1$ >0

Performance evaluation: Models are evaluated using the ROC, precision, MAE, RMSE, and confusion metrics for experiments employing the proposed system classifiers**[39].**

Table 1: SMOTE Algorithm

Step	Processes involved
	Input: Read in no. of minority samples in class T, the percentage of SMOTEs(N%), and the number of closest neighbors in k.
	Output: Synthetic minority class samples, (N/100)*T
1	Randomize minority class sample so that only a random percentage of those will be SMOTEd in the first test if N is less than 100%.
2	IF N< 100:
3	Randomize the T minority class sample
4	T← (N/100)*T
5	N←100
6	endif
7	N←int(N/100) #(*SMOTE is assumed to be integral multiple of 100*)
8	KNo. of Nearest neighbors
9	Numattrs←No. of attributes
10	Sample=[] # (*Array for original minority class samples*)
11	NewIndex=[] # (*Array for synthetic samples*)

12	For ← 1 to T
13	Compute k neighbors for I, and save to narray
14	Population(N, I, narray)
15	endfor:
16	Create function to generate synthetic samples[Population(N, I, narray)]
17	While N N≠ 0:
18	Choose random no. between 1 and k # select one of the k nearest neighbors of i
19	For attr ← 1 to numattrs:
20	Compute: diff ← sample(narray(nn[]attr] – sample[i][attr]
21	Compute: gap ← random no. between 0 and 1
22	Synthetic[newindex][attr] ← sample[i][attr] + gap * diff
23	endfor:
24	Newindex ← newindex+0
25	Endwhile
26	Return #(*End of Pseudo-code*)

4. RESULTS AND DISCUSSION

The necessary ML tools are employed to provide results and discuss findings of the proposed model. Various improvements were made to the concept and its execution to provide better and more accurate results. We presented and discussed experimental results of the proposed RF-Based classifiers using heat map, tables, ROC curve, bar and whisky charts in this section.

Figure 1: Correlation graph of the proposed system dataset

The heat map shown in Figure 1 is used to determine how closely two variables are correlated. The matrix shows that there is a link between the target variable and the pairs of the variables CA, Exam, Weighted CA, and Weighted Exam. As seen in the graph, there are both negative and positive correlations among all potential pairs of attributes.

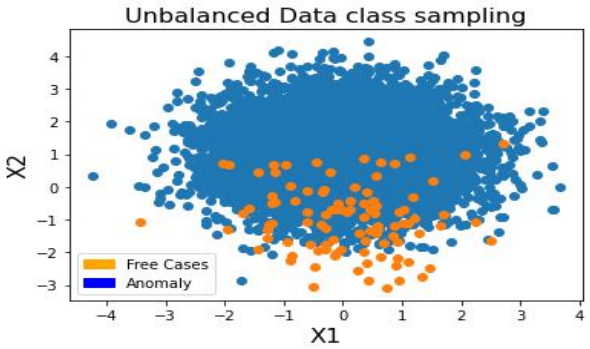

Figure 2: Unbalanced data class sampling

Figure 2 demonstrates the suggested system's imbalanced data classification of free and anomalous (CA and Exam) instances. The resulting scatter plot displays a dense concentration of points for the majority class, which is depicted in blue, and a sparse distribution of points for the minority class, which is depicted in orange. There is overlap between the two classifications, as shown.

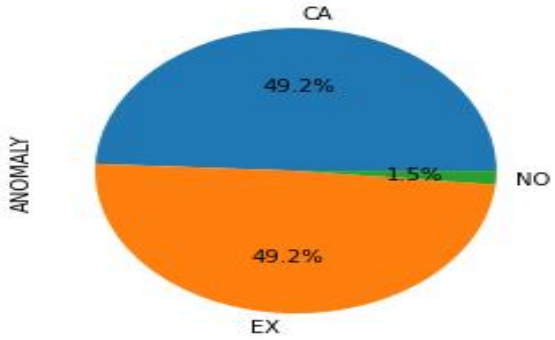

Figure 3: Imbalanced data classes

The imbalanced data class modalities of CA/Exam anomalies are shown in Figure 3 as being overrepresented in comparison to the number of free cases. From the pie chart, anomalies of the CA and exam types each made up 49.2%, whereas free cases only made up 1.5%.

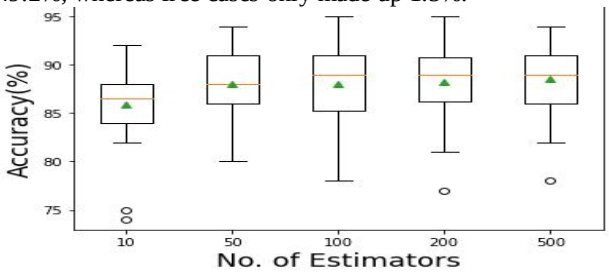

Figure 4: Adaboost classifier

Figure 4 shows the variation in the number of estimators, which ranges from 10, 50, to 500, against model detection accuracy. The adaboost classifier reported a minor change in accuracy for estimators with values between 85% and 90% exclusively.

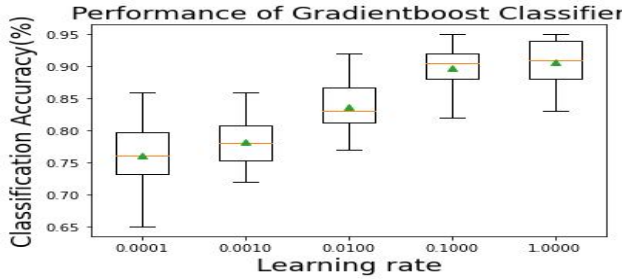

Figure 5: Learning rate of bagging

The GradientBoost plot of classification accuracy against learning rate is shown in Figure 5. The classification accuracy rises as the learning rate is increased from 0.0001, 0.0010, 0.0100, 0.1000, and 1.00 as shown in figure 5 above. The model's performance in terms of classification accuracy improves as learning rate increases.

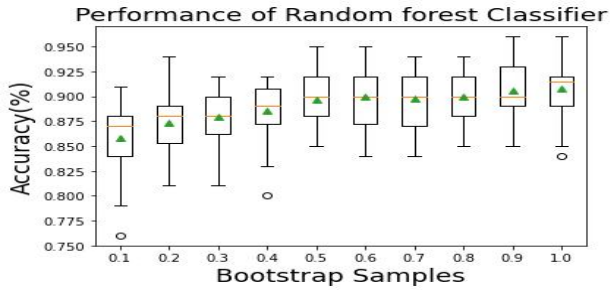

Figure 6: RF classification model

A box- and -whisker plot of RF-classification is used to visualize the variation in accuracy scores for each bootstrap sample, as shown in figure 6. A general pattern that we may observe is that as sample size increases, model performance also tends to improve.

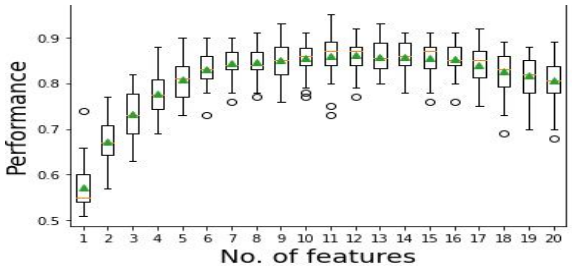

Figure 7: Performance of Adaboosting with different features

Figure 7 shows the distribution of accuracy scores for each feature selection size using the Adaboost classifier's whisker plot. Generally speaking, accuracy rises as the number of features increases until it reaches around 8 to 12 features, at which point it is roughly flat, and then performance begins to modestly decline.

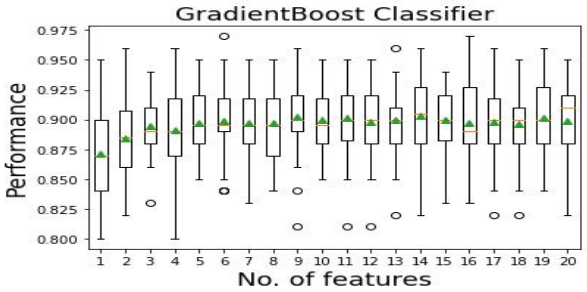

Figure 8: Metrics GradientBoost with different features

The iteration with 14 and 15 has a little greater mean accuracy than the others. From the box plot of GradientBoost classifier as shown in figure 8, we can observe the various accuracy ratings for the different sample sizes; however the sample sizes of 9 and 14 yield the best results.

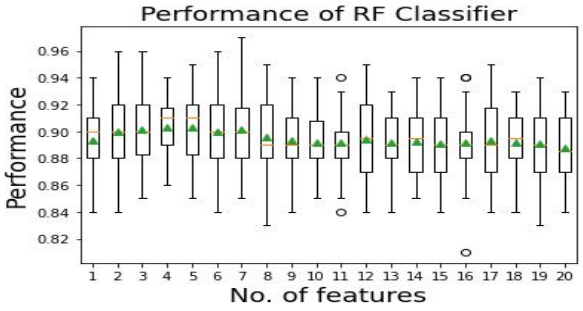

Figure 9: Performance of RF-classification with different features

The box plot of RF-classification, which was used to show the fluctuation in accuracy scores to every feature sample, is shown in Figure 9. The performance tends to increase and peak with values between three, six, and eight before declining once more when higher feature set sizes are taken into account.

Table 1: Performance of classifiers with unbalanced dataset

Metrics	RFCLF	GradBoost	AdaBoost
Mean Accuracy	0.8989	0.5264	0.3835
Mean precision	0.9101	0.4922	0.2756
Mean recall	0.8979	0.4993	0.3708
MAE	0.0633	3.4358	4.9050
MSE	0.0766	31.944	52.6616
RMSE	0.2768	5.6519	7.2568

Table 2: Performance of classifiers with SMOTEd technique

Metrics	RFCLF	GradBoost	AdaBoost
Mean Accuracy	0.9986	0.9681	0.6955
Mean precision	0.9988	0.9681	0.4352
Mean recall	0.9985	0.9687	0.4042
MAE	0.0025	0.0850	2.1591
MSE	0.0025	0.3050	12.3675
RMSE	0.0500	0.5523	3.5167

The performance of AdaBoost, RF-classification, and gradient-Boosting as shown in table 5 and 6, the SMOTed models performed better than the imbalanced class distribution in terms of accuracy, precision, recall, MAE, MSE, and RMSE. Table 1 shows the accuracy of RF-based classifiers trained using the imbalanced data class, which is somewhat lower than the accuracy of the SMOTEd 0.9986 value as shown in Table 2 with the balanced dataset.

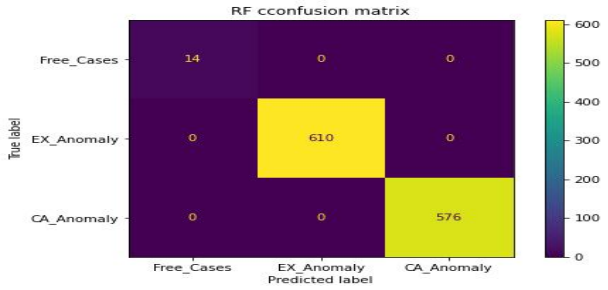

Figure 10: Confusion matrix of RF-classification

All 14 of the free cases of result abnormalities in the test samples that have been reported in the first row are correctly classified by the RF-classification. The Ex Anomaly group from the 610 Exam test sample, which is the third row of anomalies, is correctly classified by the model. In the test samples from the third row, which is related to the CA Anomaly group, the classifier accurately detected 576 CA Anomalies, as shown in figure 10.

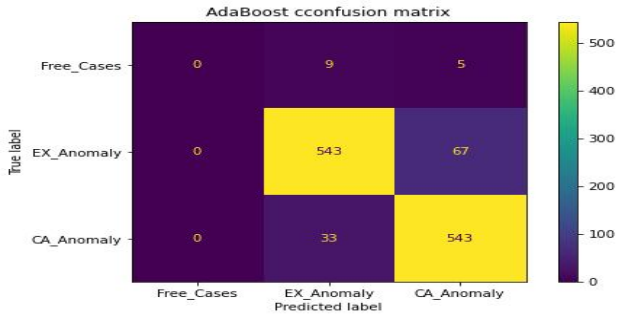

Figure 11: Confusion matrix of AdaBoost with multi-classification.

The AdaBoosting wrongly classified all the 14 free cases of result abnormalities in the test samples' as recorded in the first row. The model accurately identified 543 exam test samples (out of a total of 610 that are in the test set) and missed 67 exam test samples that are incorrectly predicted as "CA Anomaly". The classifier correctly identified 543 CA test samples from the third row and misclassified 33 CA Cases as Exam_Anomaly as shown in figure 11.

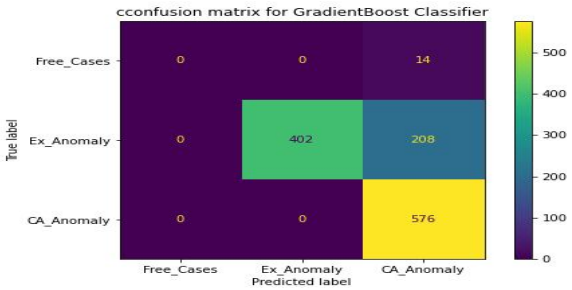

Figure 12: Confusion matrix of GradientBooost classifier

From the Free Cases group in the first row; the first box on the upper left has the value 0, while the two boxes behind it have the numbers 0 and 14. The GradientBoost misclassified all 14 of the free test samples. The model misclassified all of the free cases as "CA Anomalies." The model accurately identified 402 exam test samples (out of a total of 610 that are in the test set) and missed 208 exam anomalies test samples that are incorrectly projected as "CA Anomaly" according to the second row of the EX Anomaly group. As shown in figure 12, GradientBoost classified 576 CA test samples from the third row, which corresponds to the CA Anomaly group.

Table 3: Result anomaly

	CA	EX	WCA	WEX	DIFF	ANOMALY	AdaB	GrdCLF	RFCLF
0	28	6	70.0	10.0	60.0	EX	EX	EX	EX
1	37	44	92.0	73.0	19.0	EX	EX	CA	EX
2	6	3	15.0	5.0	10.0	EX	EX	CA	EX
3	16	18	40.0	30.0	10.0	EX	EX	CA	EX
4	25	19	62.0	32.0	30.0	EX	EX	EX	EX
5	4	32	10.0	53.0	-43.0	CA	CA	CA	CA
6	16	56	40.0	93.0	-53.0	CA	CA	CA	CA
7	40	34	100.0	57.0	43.0	EX	EX	EX	EX
8	10	12	25.0	20.0	5.0	EX	EX	CA	EX
9	40	11	100.0	18.0	82.0	EX	EX	EX	EX
10	39	52	98.0	87.0	11.0	EX	EX	CA	EX
-	-	-	-	-	-	-	-	-	-
1196	39	21	98.0	35.0	63.0	EX	EX	EX	EX
1197	27	35	68.0	58.0	10.0	EX	EX	CA	EX
1198	36	7	90.0	12.0	78.0	EX	EX	EX	EX
1199	35	54	88.0	90.0	-2	CA	CA	CA	CA

The RF-classification as shown in table 3 was able to identify the exact number of CA and exam anomalies and free cases in the result test dataset. Exam anomalies were mistakenly classified as CA anomalies by the GradientBoost classifier (a mismatch between CA and exam anomalies) is indicated by red coloring in Table 3 in Series 2-3, 8, 10, and 1197. The AdaBoost classifier was able to identify anomalies of the CA and exam types, but it was unable to recognize free cases in the test sample.

Table 4:Meantest error and standard test error of GradientBoost			
ranking	estimators	mean_test_err	std_test_err
1	5	0.19785714	0.01284722
2	500	0.15857143	0.00749149
3	15	0.16142857	0.00840614
4	5	0.17500000	0.01148424
5	200	0.18392857	0.01226826
6	10	0.19071429	0.01622545
7	5	0.20571429	0.01811950
8	5	0.22642857	0.01991692
9	15	0.24428571	0.02138686
10	20	0.27678571	0.02217164

Table 5: Mean test error and standard test error of RF			
ranking	estimators	mean_test_err	std_test_err
1	100	0.15678626	0.01666310
2	50	0.14857287	0.00305591
3	15	0.15107415	0.00387874
4	200	0.15249978	0.00842146
5	5	0.15250131	0.00963594
6	10	0.15642976	0.01167034
7	15	0.15928639	0.01378188
8	5	0.19893385	0.01386109
9	500	0.27106959	0.01635972
10	15	0.27428464	0.02229198

Table 6: Mean test error and standard test error of AdaBoosting			
ranking	estimators	MeanTestErr	std_test_err
1	10	0.24036	0.003819
2	300	0.198214	0.003819
3	1	0.203928	0.003819
4	150	0.204286	0.004394
5	3	0.205356	0.009080
6	100	0.207143	0.010509
7	2	0.237158	0.010596
8	400	0.276784	0.012267
9	500	0.276784	0.031136
10	5	0.276784	0.003819

From the analysis, we observed that the top-ranked models had a smaller parameter learning rate of mean test error and standard test error, necessitating the use of more DT trees or more leafs per decision tree.

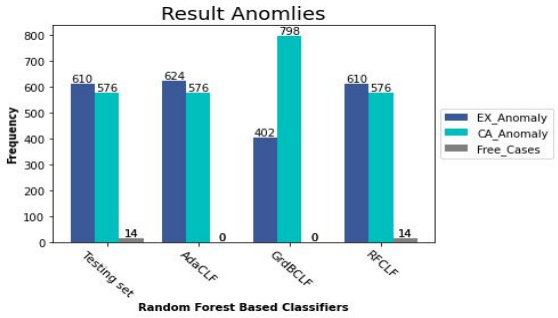

Figure 13: Summary of result anomalies

The number of detected CA, Exam abnormalities, and free cases gleaned from the testing dataset are shown in The summary graph in Figure 13. GradientBoost yielded 402 exam anomalies, 798 CA anomalies, and 0 free cases, compared to the Bagging classifier's 610 exam anomalies, 576 CA anomalies, and 14 free cases. GradientBoost yielded 402 exam anomalies, 798 CA anomalies, and 0 free cases, compared to the AdaBoost classifier's 624 exam anomalies, 576 CA anomalies, and 0 free cases. Additionally, RF-classification produced 14 free cases, 576 CA anomalies, and 610 exam anomalies. The testing dataset contains the same number of CA and exam anomalies as those record by the RFCLF.

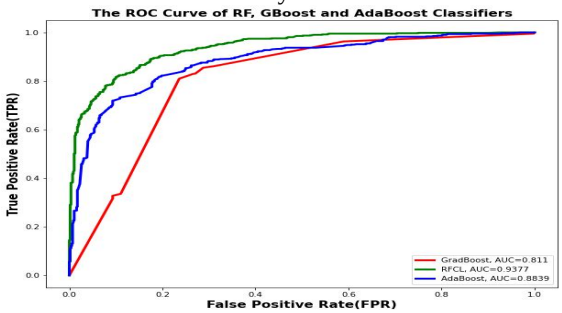

Figure 14: The ROC curve.

The receiver operating characteristic (ROC) graph in Figure 14 depicts the trade-off between true positive rate (TPR) and specificity for the RF-classification, GradientBoost, and AdaBoost classifiers (1-FPR). The RF-classification ROC curve is higher than Adaboost and both are located closer to the top-left corner of the graph. The GradientBoost curve in the ROC graph recorded AUC value of 0.811, the RF-classification produced 0.9377, and the best value was obtained by the AdaBoost curve, which had AUC value of 0.8839.

5. CONCLUSION

Machine learning is quite successful in multiple fields, among which anomaly detection is a feasible application that attracts lots of professional's attention. We propose an in-depth analysis of the use of the AdaBoost, GradientBoost, and RF-based classification techniques. Strong decision-Tree learners are automatically sought out and added to the majority class using the Grid search with log uniformly distributed function. The RF is employed

because of its precision and capacity to merge numerous decision trees in making decision. Two classes make up an unbalanced dataset, with the majority class having an unreasonably larger amount of observations than the minority class. This causes the model to produce unexpected outcomes. As a result, handling imbalanced data is necessary to guarantee the success of ML model. SMOTE allows us to raise the minority class's observations in a balanced manner, which improves the model's performance. Results of AdaBoost and RF-classification recorded the highest level of accuracy no matter how many new trees were added to the forest and reported the lowest accuracy rate in terms of conformance to the true value depending on how many trees are growing in the forest,

Acknowledgement

We thank the Nigerian Tertiary Education Trust Fund (TETfund) for sponsoring this research through an institutional-based research(IBR) grant. The sponsor played no part in the design, data gathering, or application of the findings.

References

[1] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation Forest: A New Classifier Ensemble Method," IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI), Vol. 28, Issue.10, pp.1619 – 1630, 2006

[2] L. Breiman, "Random Forests, Machine Learning," Vol.45, no.1, pp.5-32, 2001

[3] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forest and Decision Trees," International Journal of Computer Science Issues, Vol.9, no.5, pp.272-278, 2012

[4] R. R. Chaudhari, and S. P. Patil, "Intrusion Detection System: Classification, Techniques and Datasets to Implement," International Research Journal of Engineering and Technology (IRJET), Vol.04, no.02, pp.1860- 1866, 2017

[5] S. Talla, P. Venigalla, A. Shaik, and M. Vuyyuru, "Multiclass Classification Using Random Forest Classifier," International Journal of Scientific Research in Computer Science," Engineering and Information Technology(IJSRCSEIT), Vol.5, no.02, pp.493- 496, 2019

[6] V. Y. Kulkami, and P. K. Sinha, "Random Forest Classifiers: A Survey and Future Research Directions," International Journal of Advanced Computing Vol. 36, no.1, pp.1144-1153, 2013.

[7] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust Random Cut Forest Based Anomaly Detection on Streams," International Conference on Machine Learning(ICML), pp.2712-2721, 2016

[8] C. C. Aggarwal, "Outlier Analysis, Springer Publishing Company," Incorporated, 2nd Edition, pp.50-78, 2016

[9] A. Liaw, and M. Wiener, "Classification and Regression by Random Forest," R News, Vol. 2, no.3, pp.18-22, 2002.

[10] L. I. Kuncheva, "Combining pattern Classifiers: Methods and Algorithms," John and Sons, pp.23-46, 2014.

[11] G. Prashanth, GV. Prashanth., P. Jayashree, and N. Srinivasan, "Using Random Forest For Network-Based Anomaly Detection at Active Routers," IEEE-International Conference on Signal processing, Communications and Networking Madras Institute of Technology, Anna University Chennai India, pp.93-96, 2008.

[12] N. Quadrianto, and Z. Ghahramani, "A Very Simple Safe-Bayesian Random Forest," IEEE Transactions on Pattern Analysis and Machine Intelligence," Vol.37, no.6, pp.1297-1303, 2015.

[13] R. Primartha, and B. A. Tama, "Anomaly Detection using Random Forest: A Performance Revisited," 2017 International Conference on Data and Software Engineering(ICoDSE), pp.1-7, 2017.

[14] M. M. Brueing, H. P. Kriegal, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," ACM SIGMOD Record, Vol.29, no.2, pp.93-104, 2000.

[15] Salami, H. O., Ibrahim, R. S., and Yahaya, M. O.(2016). "Detecting Anomalies in Students," Results Using Decision Trees. International Journal of Modern Education and Computer Science(MECS), Vol. 8, Issue. 7, 1312–1317. doi:10.5815/ijmecs.2016.07.04.

[16] C. Callegari, S. Giordano, and M. Pagano, "An Information-theoretic Method for the Detection of Anomalies in Network Traffic," Computers & Security, Vol.70, pp..351-365, 2017.

[17] F. Palmieri, and U. Fiore, "Network Anomaly Detection through Nonlinear Analysis," Computers & Security, Vol.29, no.7, pp.737-755, 2016.

[18] A. Sabha, "Anomaly-Based Intrusion Detection using Machine Learning Algorithms - A Review Paper," International Research Journal of Engineering and Technology (IRJET), Vol.7, no.8, pp.4066- 4070, 2020.

[19] V. Jyothsna, and V. V. R. Prasad, "A Review of Anomaly based Intrusion Detection Systems," International Journal of Computer Applications, Vol.28, no.7, pp.26- 35, 2011

[20] H. R. Roplekar, and N. V. Buradkar, "Survey of Random Forest Based Network Anomaly Detection Systems," International Journal of Advanced Research in Computer and Communication Engineering, Vol.6, no.12, pp.95-98, 2017.

[21] S. Omar, A. Ngadi,. and H. H. Jebur, "Machine Learning Techniques for Anomaly Detection," International Journal of Computer Applications, Vol.79, no.2, pp.33- 37, 2013.

[22] N. P. Bhadri, F. D. C. Clement, and M. Gouvtham, "Comparing the Performance of Anomaly Detection Algorithms," International Journal of Engineering Research and Technology(IJERT), Vol.9, no.07, pp.1101-1106., 2020.

[23] A. A. Hadi, A. A. "Performance Analysis of Big Data Intrusion Detection System Over Random Forest Algorithm," International Journal of Applied Engineering Research, Vol.13, no.2, 1520-1527, 2018.
[24] S. Ziweritin, B. B. Baridam, and U. A. Okengwu, "Neural Network Model for Detection of Result Anomalies in Higher Education," Scientia Africana: An International Journal of Pure and Applied Sciences(IJPAS), Vol.19, no.2, pp.91-104, 2020.
[25] L. Felipe, B. Efe, and S. Miguel, "Categorization of Anomalies in Smart Manufacturing Systems to Support the Selection of Detection Mechanisms," Hybrid Dynamical Modeling of Electro-hydro dynamic Jet Printing, International Journal on IEEE robotics and automation, Vol.2, no.4. pp.45-60, 2017.
[26] S. B. Rahayu, N. D. Kamarudin, and Z. Zainol, "Case Study of UPNM Students Performance Classification Algorithms," International Journal of Engineering and Technology, Vol.7, pp.285–289, 2018.
[27] O. S. Hamza, S. Ruqayyah, and O. Mohammed, "Detecting Anomalies in Students Results Using Decision Trees," International Journal of Modern Education and Computer Science(MECS), pp.1312–1317., 2016 doi:10.5815/ijmecs.2016.07.04.
[28] E. I. Al-Fairouz, and M. A. Al-Hagery, "Students Performance: From Detection of Failures and Anomaly Cases to the Solutions-Based Mining Algorithms," International Journal of Engineering Research and Technology, Vol.13, no.10, pp.2895-2908, 2020.
[29] V. Shanmugarajeshwari, and R. Lawrance, "Analysis of Students' Performance Evaluation using Classification Techniques," International Conference on Computing Technologies and Intelligent Data Engineering(ICCTIDE), pp.1–7, 2016.
[30] J. H. Sharp, and I. A. Sharp, "A comparison of Student Academic Performance with Traditional, Online, and Flipped Instructional Approaches in a C# programming course," Journal of Information Technology Education: Innovations in Practice, Vol.16, no.1, pp.215–231, 2017.
[31] V. N. Uzel, S. S. Turgut, and S. A. Ozel, "Prediction of Students' Academic Success Using Data Mining Methods," in 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), pp.1–5, 2018.
[32] G. P. Zhang, "Neural Networks for Classification: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Vol.30, no.4, 451-461, 2000
[33] O. A. Aderemi, and A. A. Andronicus, "A survey of machine-learning and nature-Inspired Based Credit Card Fraud Detection Techniques," International Journal of System Assurance Engineering and Management(IJSAEM) Vol.8, no.2, pp.937-953, 2017.
[34] D. Mahapatra, "Analyzing Training Information from Random Forests for Improved Image Segmentation," IEEE Transactions on Image Processing, Vol.23, no.4, pp.1504-1512, 2014.
[35] N. M. Abdulkareem, and A. M. Abdulazeez, "Machine Learning Classification Based on Radom Forest Algorithm: A Review," International Journal of Science and Business(IJSAB), Vol.5, no.2, pp.128-142, 2021.
[36] R. Hasan, S. Palaniappan, A. R. A. Raziff, S. Mahmood, K. U. Sarker, and A. Rafi, "Student Academic Performance Prediction by using Decision Tree Algorithm," In 2018 4th International Conference on Computer and Information Sciences (ICCOINS), pp.1–5, 2018
[37] J. Han, M. Fang, S. Ye, C. Chen, Q. Wan, and X. Qian, "Using Decision Tree to Predict Response Rates of Consumer Satisfaction, Attitude, and Loyalty Surveys," Sustainability, Vol.11, no.8, pp.2306, 2019.
[38] Z. Zhou, Y. Wang, X. He, and X. Zhang, "Optimization of Random Forests Algorithm Based on ReliefF-SA," IOP Conference Series: Materials Science and Engineering, 768, pp.072065, 2020.
[39] M. A. Al-Hagery, "Classifiers' Accuracy Based on Breast Cancer Medical Data and Data Mining Techniques, International Journal of Advanced Biotechnology and Research, Vol.7, no.2, pp.760–772, 2016

AUTHOR

First: S. Ziweritin, holds HND(Computer Science) from Rivers State(Now Kenule Benson Saro-Wiwa) Polytechnic, Bori. PGD(Computer Science) and M.Sc(Computer Science) from University of Port Harcourt(UPH) respectively. He works as a lecturer in the department of Estate Management and Valuation, School of Environmental Design and Technology, Akanu Ibiam Federal Polytechnic, Unwana, Ebonyi State. His research interest revolves around: Artificial Intelligence (AI), Data Science, Machine Learning(ML), Deep machine learning(DML), Algorithms, Interactive Machine learning(IML), Database System Design and Programming. He has published in several international journals relating to AI, Data mining and Machine learning.

Second: I. A. Ibiam, Holds ND (Building Technology) from College of Technology(Now Federal Polytechnic) Nekede Owerri, BSc(Estate Management) from Imo(Now Abia) State University Uturu, MBA and MSc(Environmental Resource Management) from the University of Calabar(Unical). He is presently a Chief lecturer in the department of Estate Management and Dean, School of Environmental Design and Technology, Akanu Ibiam Federal Polytechnic, Unwana, Ebonyi State. He is an Associate member (ANIVS) of the Nigerian Institution of Estate Surveyors and Valuers (NIESV), and a registered Estate Surveyor and Valuer (RSV). He has published in several national and international journals.