University of Ibadan
Journal of Science and Logics in ICT Research

An Improved Ensemble Model Using Random Forest Branch Clustering Optimisation Approach

Ayinla, I. B. and Akinola, S. O.

University of Ibadan, Ibadan, Nigeria
ayinlab@gmail.com, akinola.olalekan@dlc.ui.edu.ng

Abstract
The world of technology is growing faster and helping organisations to repositioning their focus and vision for business. The introduction of Internet of Things (IoTs) devices has contributed in no small measure to business values and the world livelihood. The need for efficient Machine Learning Algorithms (MLAs) to drive these devices to perform to optimal or near optimal has been a serious challenge. The inadequacies of these MLAs has resulted in loss of trust and sometimes led to legal litigation against Artificial Intelligent (AI) organisations. Hence, we introduced a novel approach to improving traditional Random Forest RF, an ensemble model, which is known to be high performance classifier using branch clustering Random Forest (BCRF) technique in Decision Tree Forests (DTFs). The sensitivity, specificity and F-score values as well as extra pruning of pessimistic after Entropy and Information Gain Ratio (IGR) were used to isolate the weaker groups for model improvement. The model produced more accurate results with a better speed of execution when used on the same dataset as Naïve Bayes, RandomForest and K-nearest Neighbour.

Keywords: Decision Tree Forest, Random Forest, Classification and Prediction

1.0 Introduction

The world of technology gave birth to the generation of Artificial Intelligence which gave rise to machine learning algorithms on which IoT devices run. These devices were meant to communicate appropriately with their immediate environment, respond to reaction from their surrounding based on the pre-configured algorithms. In fact, the ultimate vision of IoT, Things was to be identifiable, self-governing, self-configurable and ability to sense and actuate predictively to event stimuli.

The journey began with research on packet switching as a medium of exchanging data and since then, the growth of Information and Communication Technologies (ICT) has been spontaneous [3]. More so, as our society began to move towards the awareness of this "monster" called internet, there is increasing use of online tools and cybersecurity threat on which IoT expected to play a major role.

According to a cybersecurity company, Helsinki, Finland-based F-Secure with regards to the research from Gartner 2020, the number of Internet of Things (IoT) devices per households will climb steeply to 500 by year 2022 from nine currently, with IoT connectivity included in the package without being optional [4].

In 2018, Mikko Hypponen, Chief Research Officer for F-Secure published a report on the IoT devices, that device without IoT facilities may not be affordable any longer because manufacturers may not be able to harvest data from them, whereas, it is this data that makes the IoT an ideal for businesses. Although, this data comes with diverse challenges or risk factors, which needs to be overcome quickly [4].

The importance of machine learning to the effectiveness of these devices cannot be overemphasised. The implications of the failure of the AI system sometimes are grievous and results in a greater loss of business values.

Ayinla, I. B. and Akinola, S. O. (2022). An Improved Ensemble Model Using Random Forest Branch Clustering Optimisation Approach, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 7 No. 2, pp. 8 - 19

In 2012, IBM lost 62 million dollars, due to the failure of an AI system built to fight cancer. The result of the performance of the system was rather disappointing because of wrong prediction of treatment to patient suffering from the disease. The algorithm recommendation was incorrect and this resulted in severe bleeding in patient. Apparently, such a sensitive and complex domain as instrument to treat life-threatening conditions is a bit too risky to fully trust it to AI because of the inconsistency in performance of these algorithms [19].

The ensemble models inferred from ensemble algorithms have proven to be more efficient as classification and prediction tools when compared with most other popular classification application [5]. The ensembles classification and regression are established techniques known for high level performance when having many diverse models in the classification forest [1, 2, 8 13]. The emphasis has been on clustering of similar classifiers with relation to their pattern, and selecting a representative from among them from each group that can lead to smaller but efficient models and more diversified ensemble.

The Traditional Random Forest (TFR) was an ensemble classifiers known as high-end performer but consumed more computing resource and required fine-tuning to enhance their predictability traits, which is the main objective of this study. However, the BCRF was developed to work on the weaknesses of TRF by introduction of a new pruning technique. The idea was to break down the branches of different models generated in the DTFs to form cluster of similar branches. The contribution of each of the cluster to the whole performance of the model was determine using confusion matrix metrics qualitatively rating the effectiveness and subsequently eliminates cluster that had less success in classification process.

The rest of this paper is organised as follows. The next section is the review of related works. Section 3 describes the proposed method of pruning. In section 4 the experimental results are discussed. Finally, section 5 provides some conclusions.

2.0 Related Works

The search for appropriate algorithms for medical health dataset to improve health services has been on for a while. The move for such a solution is the prediction of chronic kidney diseases using Decision Tree family algorithms C4.5. When the performance of the classifier was checked, it produced an appreciable improvement in terms of accuracy and speed of execution.

Manogaran and Lopez [9] made an effort by developing a Stochastic Gradient Descent (SGD) algorithm with logistic regression. The researchers used Apache Mahout to produce diagnosis model that is robust and scalable. The model achieved 81.99% and 81.52% accuracies when applied on training and testing samples respectively.

According to Fawagreh and Gaber [6] in their research work on "Resource-efficient fast prediction in healthcare data analytics: A pruned Random Forest (RF) regression approach" similar group of trees in the Decision Tree Forest (DTF) were gathered and representatives from each of the group were picked to form the final pruned model. It is understood that the models in the RF will contain different samples because of diversity in the training set used to train it. The randomisation of the DTF was seen in the technique of selecting the best node when determining the splitting point.

The CLUB-DRF method was used on three different dataset: leukemia, lung cancer, and heart disease [10] and the performance was checked against classifiers: Naive Bayes, C4.5, and Random Forest. The result of the experiment shows improvement with almost the same accuracy as the traditional RF. The CLUB-DRF method produced better accuracy, precision, recall and F-measure.

Wang *et. al.* [14] proposed a pruning algorithm of convolutional network based on optimal threshold. The algorithm adopts a technique using optimization strategy of greedy algorithm to select an optimal threshold. The algorithm performs optimisation using the sensitivity and correlation of each feature to determine their contribution either positive or negative in the model. The node in the decision tree that contributes below the expected threshold is truncated to improve the algorithm performance. The algorithm is used to test VggNet network pruning on the CIFAR-10 dataset and it generates an encouraging results with reduction in network

parameters and running time. The optimisation method using greedy algorithm was adjudged a pretty idea.

According to Özöğür-Akyüz, Otar, and Atas [11], ensemble cluster pruning can be used via convex-concave programming. The idea was to aggregate different decision tree models and relies on the degree of individual performance jettisoning the accuracy to select a group of best performance models to form the new algorithm. The algorithm was reported to have found better and approximated results to the optimum solutions. It was perfectly used to determine the best subset of models and the best one out of all the available models. The ensemble strategy has its advantages however, this proposed method trade-off accuracy for other metrics to arrive at best model.

Gao *et. al.,* [7] used rethinking pruning method for accelerating deep inference at the edge to facilitate real- time response of data mining and high- accuracy. This is necessary especially in devices that meant for speech recognition and language understanding and are having memory and storage space constraints. In tackling these challenges, network pruning algorithms have been developed and found to perform effectively well in deep neural network.

As good as they are, they often slow down devices and does increase latency time. However, rethinking pruning was introduced to alleviate these problems by suggesting an entropy-based pruning into network pruning technique without loss of accuracy. The sole aim is to reduce the information entropy of deep neural network outcomes to reducing the upper bound of the next decoding search space. The model was validated against some models. It was notably revealed that entropy-based pruning method performed better with reduction in latency time though with a little loss of accuracy.

The inconsistencies in the performance of algorithms in the above studies have rendered most MLAs classifiers unreliable and undependable. The closest study to this work has considered the clustering of each model in the DTFs for improvement. The BCRF has put forward the further breaking down of these models into branches for more scrutiny to improve the reliability of the model by digging deeper into their contributions.

3.0 Methodology

The methodology adopted in this study is best illustrated with Figure 1. The figure captures the main goal of this research work.

a) Branch Clustering
The branches that have similar order of features and attributes were grouped together to form a cluster in a DTF. In the same way, the whole process was repeated for all the DT branches in the second DTF that did not have any extra pruning mechanism induced.

b) Selection of Representatives
The representative with high performance measure in each group was selected based on the values of recall, precision, F-score and accuracy. The branch with less than 60% was ignored or trimmed, which simply means a cluster is removed. This eventually led to reduction in model size and time of computation.

c) Model Evaluation
The confusion matrix metric was used to judge the performance of each of the model selected for final model assessment. The test set was used on the new model and Naïve Bayes, RandomForest and k-Nearest Neighbour (KNN). The results of the accuracies were finally computed and examined to assert some conclusions.

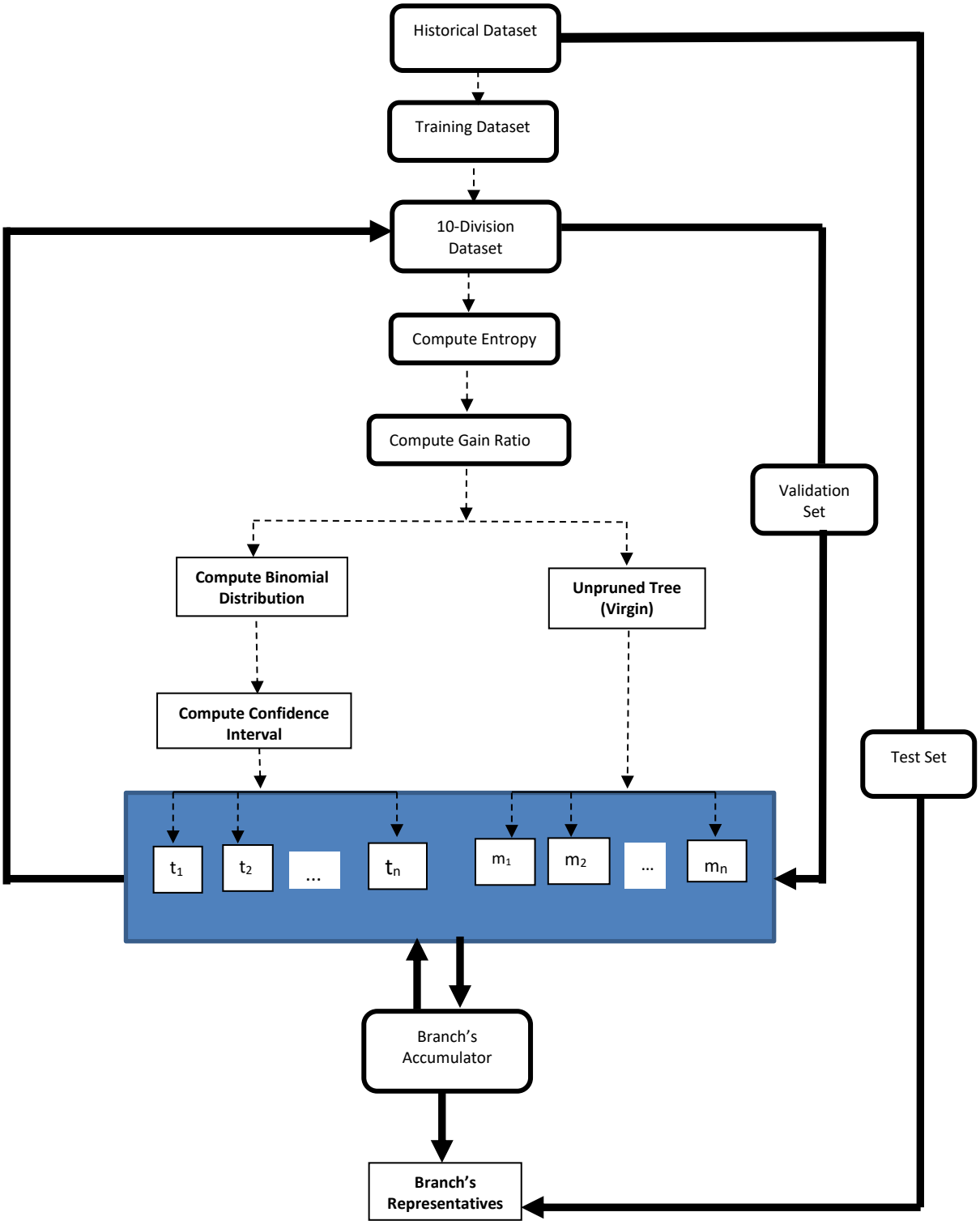

Figure 1: Decision Tree Forest Building Model CPM

d) Decision Tree Forest (DTF) Building

Two Decision Tree Forests (DTFs) were originally built using Entropy and Information Gain Ratio (IGR) before another DTF was further worked on to deny some of the Decision Trees (DTs) branches that failed performance threshold from reaching the last assessment test. The Pessimistic Pruning method was used to perform the pruning. The two DTFs were converted to rules and subsequently broken down into various distinct branches for

clustering. There were certain numbers of groups in the first DTF and another set of groups in the second DTF. The Recall (sensitivity), Precision (Specificity), F-Score (F-measure) and accuracy values of each of the branches were noted during validation and this was used to select a representative in each similar group in both DTFs.

The above features were properly cleaned to aid machine learning. Meanwhile the whole dataset was divided into two, in the ratio of 80:20% while the 80% of the dataset was used to train and build the two DTFs. The remaining 20% was used to check the effectiveness of the models. In an attempt to make sure that all features were represented, the first division; 80% dataset was partitioned into ten equal sets and leave-one technique of validation was employed by sequentially isolating a set to validate the branches of the DTs in the DTFs.

The DTFs were built on the premise of Information Gain Ratio (IGR) and Entropy principle. An average of nine DTs was built into each DTF. Beside the two principles used in training the model, the first DTF was subjected to process with pessimistic pruning technique applied on the models in DTF.

The pessimistic strategy used binomial distribution ($\varepsilon'(T, S) = \varepsilon(T,S) + \frac{|leaves(T)|}{2.|S|}$) and Confidence Interval Theory ($\varepsilon'(\text{pruned}(T, t), S) \leq \varepsilon'(T, S) + \sqrt{\frac{\dot{\varepsilon}(T,S).(1-\dot{\varepsilon}(T,S))}{|S|}}$) to determine less efficient branches in the models. The idea was to reduce the branches based on their performances. Subsequently, the algorithm began to cut down the branches of each DT from the root to the leaves to extract relevant rules. These rules represent the branches of a DT from which they were extracted. The features and attributes of the next instance in the isolated test set were retrieved from the database and compared with the features and attributes of each of the branches to find out if any branch consists of all the features and attributes. The leave of the branch under examination was checked against the target attribute of the instance. The process was recursively performed with all the instances of the test set and records of total number of checks, correct and wrong predictions were made. The precision, recall, F-Score and accuracy of the each branch were recorded against each branch.

Algorithm 1 runs through the process of building the forests from data space D, which is split into ten sets d. As stated in Figure 1, the algorithm selects both feature and attributes' identifiers from the database with the best value D_{best}, d_{best} using Shannon's entropy and Gain ratio theories. These theories are used to check the level of impurity and amount of information available in the dataset. The construction of tree T_v and A_D stops when all attributes contain the same class (pure) or there is no more feature or attribute left or no more attribute that can enhance the model.

Algorithm 1:

1. **Input**: an instance; dataset D
2. **Output** : rules *// Both pruned and unpruned rules*
3. **If** *Ds have the same class "pure" OR empty then*
4. *terminate*
5. **end if**
6. **for** *all attribute d* $\in$ *D* **do;**
7. **if** *d = spliting feature* **then**
8. *Compute Entropy*
9. *Compute information Gain Ratio*
11. **end if**
12. **end for**
13. D_{best} ← *Best attribute selected according to Shannon theories of impurity*
14. *Tree* ← *Create a decision node that tests* d_{best} *in the root //first feature is the root*
15. D_v ← *Induced children from D based on* d_{best}
16. **for** *all* D_v **do**
17. $Tree_v$ ← (D_v)*; // construct trees v using pessimistic pruning.*
18. *Induced the rule with* $Tree_v$ *and validate them*
19. $Tree_v$ ← *Virgin tree* (D_v) *without Pruning*
20. *Induced the rules from DT forests*
21. *Compute true positive, false_positive, true_negative, false_negative*
22. *Compute precision, recall, f-score and accuracy*
23. *Place similar branches from both DTFs together to form different clusters of rules*
4. **end for**
25. **for** *each rule* A_D **do;**
26. *Run filter algorithm by evaluation*
27. **end for**
28. r**eturn** *pruned rules*

e) Historical Dataset (Email log)

The email log obtained from an email server in a university in Nigeria was used in this study. The email log (EL) spanned across five years. The EL was downloaded from a Linux server for curation and cleaning purposes. There were 3,000 instances in the log with the exception of some irrelevant records. The dataset had 2,884 valid mails and 116 spammed mails when manually sorted.

f) Data Cleaning, Imputation and Transformation

The process began with the selection of relevant features to the experiment from the bulky file downloaded and filling of missing values especially the classes. The file contained many irrelevant records and features. The features selected were eleven in number based on their degree of relevancy to the operation. The SenderIP which indicates the sender‘s network (Internet Protocol Address) or the source of the massage, the sender Username (UN) on the email header, which were either acceptable or jargons, and Receive Time (time the message was delivered to the server (day/night)), which is also critical to the classification of the dataset.

Context Length was picked to examine the length or the size of the message because email’s servers were not meant to transfer or carry heavy files. They use Simple Mail Transfer Protocol (SMTP) to perform their activities. The Frequency (FRQ) of the mails is also considered, which is the number of times a similar message’s header appears. The Context Type was considered as a vital attribute, which is often in Hypertext Markup Language (HTML), plain text or multipart, Protocol Validation (PV); the email protocol must be checked to determine if it is in conformity with the policy rules. This feature is highly relevant to this study. The email server with Sender Framework Policy (SPF) needs to lookup to the records of Domain Name Server (DNS) where the email claimed to be coming from. The information from this process is compared with Internet Protocol (IP) address of the original server earlier obtained.

If the outcome matches, then the mail is passed, otherwise it is failed. The Receiver Number (RN), the structure of the receiver address on the mail header is also checked for validity and it is noted as one of the relevant features. Attached weight or attachment weight (AW), the weight of the attachment is also an important feature. Once the size is above certain threshold, it may be a subject of suspect. The result of spamAssasin; a software implemented on the server to detect spam mails was also used to improve the classification result (Target).

g) Features and Attributes of Dataset

The attributes of each of the feature from the obtained instances and as used during classification and prediction processes are as shown in Table 1:

Table 1: Features and Attributes of Dataset

S/N	Feature	Attributes
1.	SenderIP (SIP)	valid/invalid
2.	User name (UN)	valid/invalid
3.	Receive time (RT)	night/day
4.	Context Length (CL)	large/medium /small
5.	Frequency (FRQ)	frequent/seld om/less
6.	Context Type (CT)	html/plaintext /multipart
7.	Protocol Validation (PV)	legal/illegal
8.	Receiver Number (RN)	valid/invalid
9.	Attachment weight (AW)	heavy/light
10.	Server IP (SIP)	valid/invalid
11.	Class (Target)	spam/normal

h) Algorithm Training

The dataset was divided into two major divisions. The first set was used for training the model while the second partition was used to evaluate the performance of the new model against the selected existing models. To train the dataset, 90% of 2,400 (2,160) of the valid mails and 80% (94) of the spam mails were selected to avoid data from being skewed. The 2,254 total instances were eventually used for training while 746 instances were used to evaluate the effectiveness of the models. Again, the process of training the ensemble model involves partitioning the training set further into 10 sets. A group from the ten set was sequentially selected to validate the remaining nine DTs in the DTF.

i) Implementation Environment

The BCRF model was developed on a Linux Operating System (Ubuntu) using Java Programming Language on NetBeans Integrated Development Environment. The Dataset was stored and retrieved form MySQL Relational Database with 9 tables. The other classifiers (K-NN, Naïve Bayes, RF) were implemented on WEKA 3.8.4 version 2019 from The University of Waikato, Hamilton, New Zealand.

j) Performance Evaluation Metrics

According to report published by Lawtomated [17] titled, "4 Things you need to know about AI: Accuracy, Precision, Recall and F1-Score" in 2019 that accuracy alone is not enough metric to check performance of a model. In fact, accuracy is just one of the four main metrics to determine the potential of a model.

These metrics are collectively referred to as confusion matrix, which is the most popular technique used to measure classification algorithm's performance. The principle is that the number of correct predictions of samples is plotted against the number of incorrect predictions from the same samples. In a situation where the classifier is handling binary operation, it considered the operation in terms of true or false and positive or negative. It is very easy to explain vividly the performance of a model using these principles.

The four metrics under the confusion matrix can further be explained as follows:

1. **The Sensitivity:**

This metric reveals how many true positive are perfectly predicted out of all the actual positive outcomes from the samples. In other term it is referred to as Recall.

It can be mathematically explained as the true positive predicted divided by the sum of number of True Positive samples predicted and False Negative predicted samples.

$$Recall = \frac{TP}{TP+FN} \qquad \text{(i)}$$

However, if the value of sensitivity is high, it means most positive samples were actually classified correctly as positive samples, that is, it is the combination of True Positive and False Negative (TP+FN). The implication is that, there will be higher number of sample of False Positive (FP) measurement. Similarly, when the sensitivity is low there is bound to be high number of False Negative (FN). These were positive samples that were wrongly labeled or classified as negative [12].

2. **The Specificity:**

The specificity is the number of positive samples predicted that were actually belong to the positive group predicted. It is a metric that wants to figure out how many of the samples classified by the algorithm as being positive samples are actually positive samples among all the assumed positive samples predicted.

The mathematical explanation would be the number of true positive samples predicted divided by the sum of true positive and false positive predicted.

In a simply way, it is the number of correct samples predicted as true positive over total samples predicted as true positive samples; both correctly

predicted and the wrong predicted as true positive.

$$Precision = \frac{TP}{TP+FP} \quad \text{(ii)}$$

It checks how precise a model is when predicting true positive samples. In domain like medical high specificity is cherished since there will still be further examinations before final conclusion.

It is important to measure the rate of false positive during classification evaluation, that is, how many negative outcomes were wrongly predicted as positive or vice visa [12, 15, 16].

This is to check number of irrelevant samples in a search.

$$\text{FPR} = \frac{FP}{TN+FP} \quad \text{(iii)}$$

3. The F-Score

The above two metrics, Sensitivity and Specificity are always operating in an inverse order such that when one increase the other will is forced to decrease. However, F-score has provided a single way of representing the two metrics. It is introduced to find a balance point in between the two metric parameters, which is generally called the F1-score. It is affirmed that the higher the result of F-score, the more accurate a model is and the lower it is, the less accurate a model [12, 15, 16].

The F-score can be mathematically as explained as follows:

$$F - Score = \frac{2*Recall*Precision}{Recall+Precision} \quad \text{(iv)}$$

4. The Accuracy

The quality of a model is not perfectly determined by the accuracy alone even though is a critical performance's metric. In some situations, accuracy may be not be good and yet perfect for a domain area because of the result of one other previous metrics (Sensitivity, specificity). At some points, decisions are made on either of these metrics to justify the choice of a model. Recall could be prioritised over precision especially, when there is a high cost associated to FN**.**

Therefore, accuracy is the ratio of those samples correctly predicted, that is, the results of True Positive and True Negative to the entire data samples.

$$\text{Accuracy} = \frac{TP+FP}{TP+FP+TN+FN} \quad \text{(v)}$$

where:

- Positive (P): Actual is positive
- Negative (N): Actual is not positive
- True Positive (TP): Actual is positive, and is predicted to be positive
- False Negative (FN): Actual is positive, but is predicted negative
- True Negative (TN): Actual is negative, and is predicted to be negative
- False Positive (FP): Actual is negative, but is predicted positive [18].

4.0 Results and Discussion

4.1 The Sensitivity and Specificity Measurements

The sensitivity and specificity results on the chosen models in Figure 2 reveal the nature of how pure is the classification carried out on the sample data. From the Figure 2 Branch Cluster RandomForest (BCRF) had 0.995885 result for Sensitivity and 0.99316 for Specificity. The closest model in performance to BCRF in this metric was Random Forest with 0.994505 and 0.990424 for Sensitivity and Specificity, respectively.

4.2 The F-Measurement

Figure 3 shows the results of the F-measure which is the factor of both sensitivity and specificity. This is an attempt to find a balance point from both parameters. BCFR had 0.994521 while the closest model that is RandomForest had 0.992461.

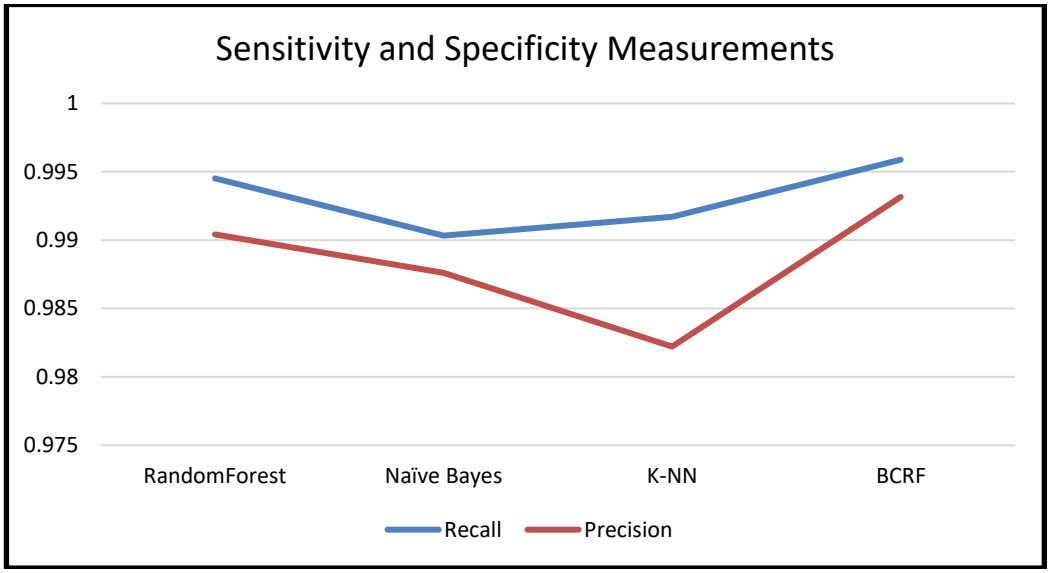

Figure 2: Models Sensitivity and Specificity Measurements

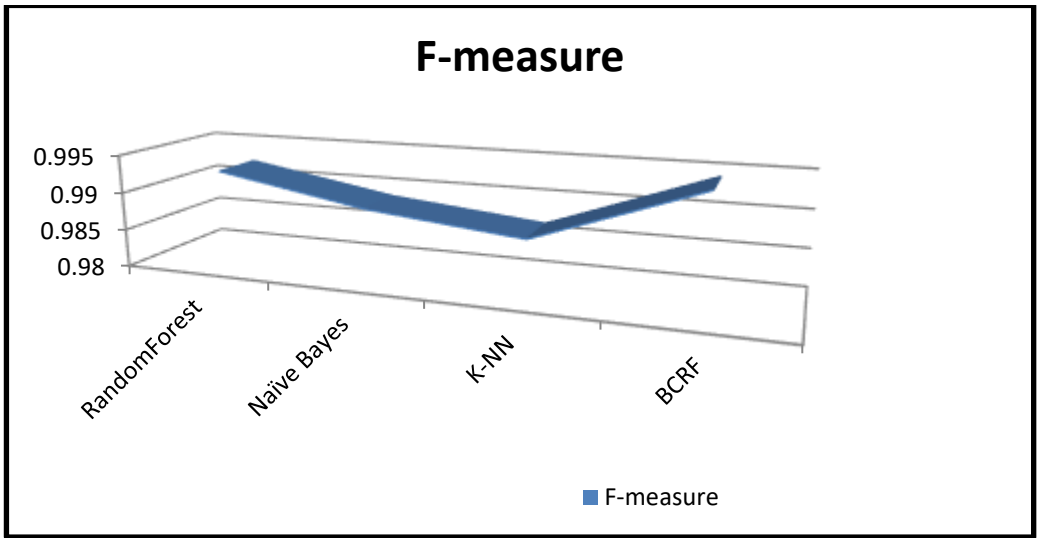

Figure 3: The models F-Measure values

4.3 Time of Execution

Figure 3 shows the time it took the models to learn and predict the outcome of the prediction assignment. The BCRF model had 0.101second to perform classification and prediction of the email set, which is relatively higher than other models except traditional Random Forest, which had the highest time of 0.11second.

4.4 Percentage of Prediction of Accuracy

The BCRF model had highest accuracy of 98.93% when compared with other models' accuracies. The Random Forest model also performed well with 98.53%, which was the nearest result to BCRF as shown in Table 1 and illustrated in Figure 4 although other models also performed fairly good.

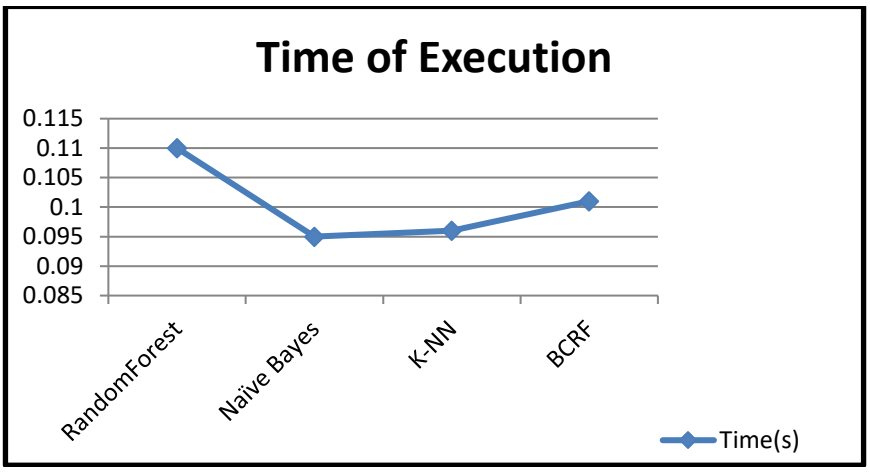

Figure 3: Execution Time(s)

Table 1: Percentage of Prediction of Accuracy

Model	TP	FP	TN	FN	Accuracy $\frac{TP+FP}{TP+FP+TN+FN}$
RandomForest	724	7	11	4	98.53
Naïve Bayes	717	9	13	7	97.86
K-NN	718	13	9	6	97.45
BCRF	726	5	12	3	98.93

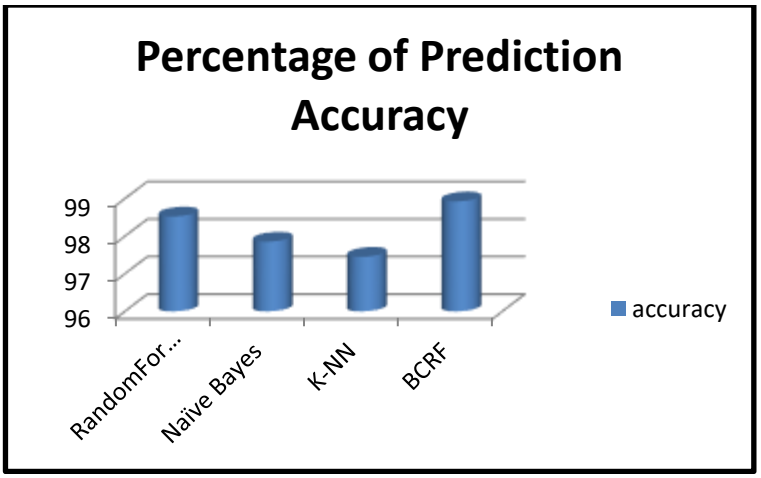

Figure 4: Percentage of Accuracy

4.5 Discussion of Results

The developed Branch Clustering Random Forest (BCRF) model was evaluated together with other standard models. The same training dataset and test subjected to the models even though they employed different approaches for learning. The training data set was made to train the model along with the classes of each of the instance as outlined in Section 3.0. The BCRF generates eighteen pruned forests having equal sizes. It breaks and converts each Decision Tree (DT) in each DTF to rules where each complete antecedent and the class of each rule are considered as a branch.

The result of the performance of each branch is kept from the DTFs. Subsequently, the procedure groups all similar rules with respect to their performance during validation and a representative is chosen from them, based on the group that achieved the highest performance in terms of accuracy during construction.

As presented in Figure 2, there is a slight improvement in BCRF result on sensitivity and specificity. This, as a key performance indicator denotes how the model classified unknown cases and how pure is the classification exercises' results after the process. In a case where the sensitivity is high and specificity is low; it implies that most of the positives is correctly classified (low False Negative, FN) but there are a lot of false positives. When the sensitivity is low and specificity is high; it means that a lot of missed positives exist (high FN) but those that were predicted as positives are indeed positive (low False Positives, FP).

The result in Figure 3 demonstrates another key performance measurement with respect to balancing both results in Figure 2 instead of condemning the outcome therein or concludes on the performance of the model without reinforcing the conclusion with other indicators. The BCRF shows a better result, which re-emphasised an improved performance in handling classification processes.

The outcome of the metric in Table 1 and Figure 4 with respect to the performance accuracy of the model in prediction shows that BCRF outperformed all the models. The approached used was rewarded with a slight improvement in the performance of the BCRF when compared with other models. It is obvious that BCRF showed superiority over others in all cases including the time of execution as depicted in Figure 3.

4.6. Limitation of the Results

The experiment was done on categorical dataset. The performance of this model will be best imagined on a non-categorical dataset.

5.0 Conclusion

The strong ability of IoT devices to react appropriately to the event in their immediate environment is a factor of the performance of the algorithm driving them. The accuracy of these algorithms is a major attribute of IoTs to differentiate between objects. One major decision that motivates this study is the desire to improve classification and prediction algorithm in order to enhance the performance of IoT devices.

We presented in this paper a Branch Clustering Random Forest (BCRF) prediction technique that is more accurate than the most standard classifiers used in this research, including the traditional Random Forest model. To achieve this objective, we empirically validate the principle of diversity in ensemble approach that often produced better performance.

The improvement of this algorithm might have been as a result of extra pruning procedures of Shannon theory and pessimistic approach in the ensemble process and the strategy employed in data partitioning to avoid skewed dataset for training. The model had small size in terms of the width and height of tree, which enhanced understandability and reduced computational time of the model. The performance was as good as other models.

As future work, the implementation of this approach using regression mechanism will be a good prospect in order to examine if the algorithm will perform consistently well or even better than categorical datasets.

References

[1] Adeva J. J. G, Beresi U, Calvo R (2005) Accuracy and diversity in ensembles of text categorisers. *CLEI Electron J* 9(1):1–2

[2] Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. *Inf Fusion* 6(1):5–20.
[3] David M. Yates (1997), Turing's legacy : a history of computing at the National Physical Laboratory 1945-1995. National Museum of Science and Industry, 1997, pp. 126–146, ISBN 0901805947 https://www.worldcat.org/title/turings-legacy-a-history-of-computing-at-the-national-physical-laboratory-1945-1995/oclc/40624091.Retrieved 19 May 2015.
[4] David Reinse, John Gantz and John Rydning (2018). The Digitlaization of the World from Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
[5] Fawagreh K, Gaber M. M, Elyan E (2014). Random forests: from early developments to recent advance-ments. *Syst Sci Control Eng* Open Access J 2(1):602–609
[6] Fawagreh, K., and Gaber, M. M. (2020). Resource-efficient fast prediction in healthcare data analytics: A pruned Random Forest regression approach. *Computing*, 1-12.
[7] Gao, D., He, X., Zhou, Z., Tong, Y., Xu, K., & Thiele, L. (2020). Rethinking Pruning for Accelerating Deep Inference At the Edge. Paper presented at the *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.New York United States.
[8] Kuncheva L. I,Whitaker C. J. (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach Learn* 51(2):181–207
[9] Manogaran G, Lopez D (2018) Health data analytics using scalable logistic regression with stochastic gradient descent. *Int J Adv Intell Paradig* 10(1–2):118–132
[10] Nagarajan V. R, Kumar V. (2018) An optimized sub group partition based healthcare data mining in big data. *Int J Innov Res Sci Technol* 4(10):79–85
[11] Özöğür-Akyüz, S., Otar, B. Ç., & Atas, P. K. (2020). Ensemble cluster pruning via convex-concave programming. *Computational Intelligence*, *36*(1), 297-319.
[12] Shruti Saxena Saxena (2018), Precision vs Recall https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488
[13] Tang E. K., Suganthan P. N. and Yao X. (2006) An analysis of diversity measures. *Mach Learn* 65(1):247–271
[14] Wang, J., Liu, L., & Pan, X. (2020). Pruning algorithm of convolutional neural network based on optimal threshold. Paper presented at the *Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence*. Chengdu China.
[15] Nillsf, (2020), Confusion Matrix, Accuracy, Recall and Precision. https://nillsf.com/index.php/2020/05/23/confusion-matrix-accuracy-recall-precision-false-positive-rate-and-f-scores-explained/
[16] Jason Brownlee, (2020). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/
[17] Lawtomated, (2019), 4 Things you need to know about AI: Accuracy, Precision, Recall and F1-Score, https://lawtomated.com/accuracy-precision-recall-and-f1-scores-for-lawyers/
[18] Sarang, N. (2019). Understanding confusion matrix**,** https://www.sarangnarkhede.com/2019/06/29/confusion-matrix/ Accessed 20th January, 2020.
[19] Angela Chen (2018) IBM's Watson gave unsafe recommendations for treating cancer. https://www.theverge.com/2018/7/26/17619382/ibms-watson-cancer-ai-healthcare-science accessed 01 December, 2020.