Article

eGAP: An Evolutionary Game Theoretic Approach to Random Forest Pruning

Khaled Fawagreh 1 and Mohamed Medhat Gaber 2,3,*

1 Department of Information Technology, Prince Mohammad Bin Fahd University, Dhahran 34754, Saudi Arabia; kfawagreh@pmu.edu.sa
2 School of Computing and Digital Technology, Birmingham City University, Birmingham B5 5JU, UK
3 Faculty of Computer Science and Engineering, Galala University, Attaka 43511, Egypt
* Correspondence: mohamed.gaber@bcu.ac.uk or mohamed.gaber@gu.edu.eg

Received: 12 October 2020; Accepted: 24 November 2020; Published: 28 November 2020

Abstract: To make healthcare available and easily accessible, the Internet of Things (IoT), which paved the way to the construction of smart cities, marked the birth of many smart applications in numerous areas, including healthcare. As a result, smart healthcare applications have been and are being developed to provide, using mobile and electronic technology, higher diagnosis quality of the diseases, better treatment of the patients, and improved quality of lives. Since smart healthcare applications that are mainly concerned with the prediction of healthcare data (like diseases for example) rely on predictive healthcare data analytics, it is imperative for such predictive healthcare data analytics to be as accurate as possible. In this paper, we will exploit supervised machine learning methods in classification and regression to improve the performance of the traditional Random Forest on healthcare datasets, both in terms of accuracy and classification/regression speed, in order to produce an effective and efficient smart healthcare application, which we have termed eGAP. eGAP uses the evolutionary game theoretic approach replicator dynamics to evolve a Random Forest ensemble. Trees of high resemblance in an initial Random Forest are clustered, and then clusters grow and shrink by adding and removing trees using replicator dynamics, according to the predictive accuracy of each subforest represented by a cluster of trees. All clusters have an initial number of trees that is equal to the number of trees in the smallest cluster. Cluster growth is performed using trees that are not initially sampled. The speed and accuracy of the proposed method have been demonstrated by an experimental study on 10 classification and 10 regression medical datasets.

Keywords: machine learning; random forests; ensemble pruning

1. Introduction

The advent of the Internet of Things (IoT) led to the construction of smart cities [1], which were mainly developed to provide high degree of information technology, and a comprehensive application of information resources. Consequently, many smart applications have come into existence in many areas including but not limited to smart energy [2], smart education [3], smart transportation [4], and smart healthcare [5]. For the latter area, which is the relevant one to the research conducted in this paper, many smart healthcare applications were developed as outlined in [6]. For such applications that are concerned with making predictions on new data based on historical data, it is imperative for such predictions to be as accurate as possible to reduce costs of treatment, avoid preventable diseases, predict outbreaks of epidemics, avoid preventable deaths, and improve the quality of life in general.

Random Forest (RF) is an effective tree-based ensemble machine learning approach for both classification and regression. RF uses bagging and randomisation heuristics to produce a diverse ensemble. Given its early success showing consistent predictive effectiveness in both classification and
regression, many attempts to further improve its performance have been made [7]. Among the methods that have been developed to build a more effective and faster RF is CLUB-DRF [8]. CLUB-DRF is able to find a subset of trees from a typically constructed RF that, in most cases, provides an even higher classification accuracy with only 5% of the trees used in RF. Given its multi-fold speed up in the inference time, makes it suitable for critical applications (e.g., healthcare). Thus, more recent work reported successful adoption of CLUB-DRF on regression problems in the area of healthcare [9].

CLUB-DRF uses clustering of trees based on similarity of prediction. Consequently, the best performing tree is selected from each cluster, forming a pruned RF, where the number of trees is equal to the number of clusters. Despite its success, the use of one tree per cluster may not be ideal, as some clusters may well include trees that are more accurate. However, the choice of a number of trees from each cluster is a computationally hard optimisation problem. In its worst case scenario, it can reach $2^t$, where $t$ is the number of trees in the original RF.

To overcome this problem, in this paper, we propose using a game theoretic approach, namely, replicator dynamics to optimise the choice of a different number of trees per cluster. We termed the new method evolutionary game theoretic approach to random forest pruning (eGAP). eGAP uses replicator dynamics heuristically to allow more trees from clusters that typically show higher predictive performance (e.g., accuracy in classification problems) to join the ensemble. An initial setting where a random selection of trees per cluster is used. The held-out trees are used for growing clusters, as per the replicator dynamics process. The other way around is also correct, where shrinking clusters put their trees to the held-out set of trees.

The paper gives a detailed account to the novel method, eGAP, showing experimentally the superiority of the method when compared to the traditional RF, and some other variations. As such, the contributions of the paper can be summarised as follows.

1. A novel evolutionary game-theoretic approach for pruning Random Forests; and
2. A faster Random Forest inference achieved through pruning; and
3. A thorough experimental study on 20 medical datasets in both classification and regression problems.

This paper is organised as follows. Section 2 presents related work in the area of smart healthcare applications. An overview of Random Forest is given in Section 3. In Section 4, an overview and detailed description of the proposed new method are given. Experiments and results are presented in Section 5. A discussion section then follows in Section 6. The paper is finally concluded with a summary and pointers to future work in Section 7.

2. Related Work

In the quest to realise smart cities, a number of predictive smart healthcare applications were developed as outlined in [6]. A prediction of inpatient violence incidents was investigated in [10] using recurrent neural network, convolutional neural network, Naïve Bayes, support vector machine, and decision tree. A study of the prediction of type 2 diabetes and hypertension was conducted by [11] using density-based spatial clustering and synthetic minority over-sampling. The authors in [12] investigated the prediction of biochemical recurrences in patients treated by stereotactic body radiation therapy using prostate clinical outlook. Using Kruskal–Wallist test, regression model, Cuckoo search optimisation algorithm, and radial basis function neural network, a model was developed in [13] to determine whether the differences of tuberculosis prevalence rates for different income groups are statistically significant or not.

For better classification and prediction performance for determining severity stage in chronic kidney disease, ref. [14] used Probabilistic Neural Networks (PNNs). In [15], a machine learning analysis, known as KNIME, was used to analyse MRI-derived texture features to predict placenta accreta spectrum in patients with placenta previa. An online Stochastic Gradient Descent (SGD) algorithm with logistic regression is implemented by [16] using Apache Mahout to develop the best scalable diagnosis model. Finally, using three different classifiers: Naive Bayes, C4.5, and Random Forest, ref. [17] investigated
the prediction of three diseases, namely, leukemia, lung cancer, and heart disease. Aligned with the work reported in this paper, in [18], the authors proposed a framework for smart healthcare using machine learning and Internet of Things. However, only off-the-shelf shallow learning methods were used including Random Forests. Inspired by this work [18], and other recent work reported in the smart healthcare domain, the research reported in this paper attempts to bring both efficiency and effectiveness to Random Forests, enabling it to contribute to near real-time decision making in smart healthcare. For comprehensive reviews of the adoption of data mining methods in smart applications including healthcare, the reader is referred to [19,20].

As can be seen, numerous attempts to use computer aided diagnosis using machine learning over the last couple of decades. However, the work on smart healthcare has seen a great deal of interest with the increasing power of handheld devices, and particularly smartphones. The work reported in this paper is an attempt in the same direction, where we apply a novel method for pruning Random Forests.

3. Random Forest: An Overview

Random Forest (RF) is an ensemble learning method used for classification and regression. Developed by Breiman [21] almost two decades ago, the method combines Breiman’s bagging sampling approach [22], and the random selection of features, introduced independently by [23,24] and Amit and Geman [25], in order to construct a collection of decision trees with controlled variation. Using bagging, each decision tree in the ensemble is constructed using a sample with replacement from the training data. Statistically, the sample is likely to have about 64% of instances appearing at least once in the sample. Instances in the sample are referred to as in-bag-instances, and the remaining instances (about 36%), are referred to as out-of-bag (OOB) instances.

To enhance diversity, at each node in all trees, a best split feature is selected using a goodness measure (e.g., Gini index) from a set of randomly selected features (typically \(\sqrt{n}\), where \(n\) is the total number of features). Each tree is grown to the largest extent possible and is unpruned. Typically, a maximum depth is usually allowed to prevent trees from growing out of memory in high dimensional datasets.

Having presented a high level overview of Random Forest, the following section discusses the proposed pruning method.

4. eGAP: An Evolutionary Game Theoretic Approach to Random Forest Pruning

In this section, a novel approach for pruning the classical RF is presented. It combines two techniques that we have used before to improve the performance of RF: clustering and replicator dynamics. These techniques are discussed next.

4.1. Clustering

Clustering was the main technique used in [8,9] to extreme pruning of random forests, and in [26], replicator dynamics was employed on a diversified random forest with subforests produced by randomised subspaces [27], to evolve subforests by allowing those with better performance to grow and those with lower performance to shrink. The use of replicator dynamics allowed subspaces of features that interact better for accurate classification to have more trees, and those subspaces that have features that are not interacting well for classification, in comparison to other subspaces, to have fewer trees.

The clustering technique utilises a well established principle in ensemble classification and regression which is that ensembles tend to perform better when the individual classifiers in the ensemble exhibit a high level of diversity [28–31]. By clustering the trees in the original ensemble (which we refer to as parentRF) into groups of similar trees, representatives are selected from each group. Consequently, many redundant trees are eliminated yielding a much smaller ensemble than
the original one. This technique has been used in two of our previously published articles [8,9], and showed notable boost in predictive performance metrics in both classification and regression.

4.2. Replicator Dynamics

Replicator Dynamics (RD) by [32] is a simple model of evolution used extensively in evolutionary game theory [33–39], and hence, explains the term eGAP, we coined for our proposed pruning method. It provides an effective way to represent selection among a population of diverse types. To illustrate how it works, assume that selection occurs between periods after dividing time into discrete intervals. The proportion of each type in the next period is given by the replicator equation as a function of the type’s payoff and its current proportion in the population. Types that score above the average payoff increase in proportion, while types that score below the average payoff decrease in proportion. The amount of increase or decrease depends on a type’s proportion in the current population and on its relative payoff.

The most general continuous form of RD is given by the differential equation in Equation (1).

$$\dot{x}_i = x_i [f_i(x) - \phi(x)] \quad (1)$$

such that

$$\phi(x) = \sum_{j=1}^{n} x_j f_j(x) \quad (2)$$

where the proportion of type \(i\) in the population is given by \(x_i\), the distribution of types vector in the population is given by \(x = (x_1, ..., x_n)\), the fitness of type \(i\), which depends on the population, is given by \(f_i(x)\), and \(\phi(x)\) is the average population fitness which is calculated as the weighted average of the fitness of the \(n\) types in the population.

4.3. eGAP Algorithm

By incorporating clustering and replicator dynamics, eGAP produces a pruned RF ensemble using the following procedure. First, a random forest of a large enough number of trees (e.g., 1000 trees) called parentRF is built. Trees in this forest are then clustered into groups of similar trees using an efficient clustering method (e.g., \(k\)-means). For clustering purposes, each tree is represented as an ordered list of classification outputs for classification tasks. The dissimilarity between any two trees is calculated using the Hamming distance between the two ordered lists. In regression, each tree is represented by the regressed values, forming a vector of real numbers. Thus, the dissimilarity between any two trees is calculated using the Euclidean or Manhattan distance between their respective vectors. The number of trees in the smallest cluster is then used to determine how many trees to draw from each cluster, forming working clusters that are all of the same initial size. The remaining trees (if any) are kept in the idle clusters, hence, for each working cluster, there is an idle one. Next, RD on the working clusters is applied—such that no working cluster can grow greater than its original cluster size, or shrink to be 0. Using RD terminology, trees in each working cluster essentially act as a type and the working cluster’s size is analogous to the proportion of each type. The working cluster’s accuracy is analogous to the type’s payoff, and the average accuracy of the working clusters is analogous to the average payoff. The discrete intervals mentioned above correspond to loop iterations. At each iteration, the performance of the trees in the working cluster being processed is compared with the performance of the trees in all the working clusters. If it is better, then the size of the working cluster grows by adding to it the best performing tree from the corresponding idle cluster, otherwise, the size shrinks by removing from it the worst performing tree and adding it to the corresponding idle cluster. The performance of the trees in the working clusters is determined by percentage accuracy for classification and R Squared for regression. Figure 1 demonstrates the main steps involved in the production of eGAP.
Putting it all together, the eGAP algorithm can be summarised as follows:

- An RF of 1000 trees is constructed.
- Trees in the forest are clustered into \( k \) initial clusters (where \( k \) is a multiple of 5 in the range 5–50).
- The number of trees in the smallest cluster is determined (let us refer to it as \( \text{minTrees} \)).
- This number is used to form the working clusters by drawing \( \text{minTrees} \) from each initial cluster.
- After drawing the trees, the initial clusters form the idle clusters.
- RD is then applied on the working clusters such that no cluster can grow greater than its original size, or shrink to be 0.
- Every time a tree is added to a working cluster, the best performing tree in the corresponding idle cluster is chosen and added to the working cluster.
- Every time a tree is removed from a working cluster, the worst performing tree is removed and placed in the corresponding idle cluster.
- After RD is applied on the working clusters, trees in these clusters are used to populate the ordered list \( T_{\text{eGAP}} \) which represents eGAP.

More formally, the above algorithm is outlined in Algorithm 1 where \( T \) refers to the training dataset, \( S \) refers to the size of the parentRF to be created, \( k \) refers to the number of clusters to be created, and \( RD\text{Iterations} \) refers to the number of RD iterations that will be applied on the working/active clusters.

The cost of applying eGAP has two components (1) applying a clustering algorithm on a relatively small dataset (each instance represents a tree/classifier in the ensemble); and (2) the cost of re-evaluating the ensemble after each iteration of the replicator dynamics process. Both components are cost efficient, as no re-training is required.

Having discussed the proposed method for pruning Random Forests (eGAP), the following section provides a detailed account of the thorough experimental study conducted to validate the effectiveness of the method.
Algorithm 1 eGAP algorithm

\[\text{Algorithm 1 eGAP algorithm}\]

\{User Settings\}
\text{input } T, S, k, RD\text{Iterations}
\{Process\}
Create an empty super ordered list \textit{AllPredictions}
Create an empty ordered list \textit{T}\textsubscript{rf} to represent parentRF
Create an empty ordered list \textit{T}\textsubscript{eGAP} to represent eGAP
Using the traditional Random Forest Algorithm, create \textit{T}_{parentRF} of size \(S\)
For each tree in \textit{T}_{parentRF}, find its predictions on \(T\) and update \textit{AllPredictions}
\textbf{for} \(i = 1 \rightarrow S\) \textbf{do}
\textit{AllPredictions} = \textit{AllPredictions} \cup \text{FindPredictions}(\textit{T}_{parentRF}.\text{tree}(i), T)
\textbf{end for}
Using K-means, cluster \textit{AllPredictions} into a set of initial \(k\) clusters:
\textit{cluster}_1 \ldots \textit{cluster}_k
From the smallest cluster, find its size \textit{minSize}
Create working clusters (\textit{wkClusters}), each has size \textit{minSize}
Add \textit{minSize} trees from initial clusters to \textit{wkClusters}
Create idle clusters \textit{idleClusters}
Add the remaining trees in each initial cluster to \textit{idleClusters}
\textbf{for} \(i = 1 \rightarrow \text{RD}\text{Iterations}\) \textbf{do}
\textbf{for} \(j = 1 \rightarrow \text{wkClusters}\) \textbf{do}
\textit{wkClustersPerformance} ← \text{calculateWkClustersPerformance(\textit{wkClusters})}
\textit{wkClusterPerformance} ← \text{calculateWkClusterPerformance(\textit{wkClusters}(j))}
\textbf{if} \textit{wkClusterPerformance} \geq \textit{wkClustersPerformance} \textbf{then}
\text{Remove the best performing tree from \textit{idleClusters}(j)}
\text{Add it to \textit{wkClusters}(j)}
\textbf{else}
\text{Remove the worst performing tree from \textit{wkClusters}(j)}
\text{Add it to \textit{idleClusters}(j)}
\textbf{end if}
\textbf{end for}
\textbf{end for}
Use the trees in \textit{wkClusters} after applying RD to populate \textit{T}_{eGAP}
\textbf{for} \(i = 1 \rightarrow \text{wkClusters}\) \textbf{do}
\text{Get the next tree from \textit{wkClusters}(i)}
\text{Add it to \textit{T}_{eGAP}}
\textbf{end for}
\{Output\}
An ordered list of trees \textit{T}_{eGAP}

5. Experimental Study and Results

This sections serves as the experimental part of the paper and includes detailed description of the experiments conducted and results. Before doing so, the required setup of the experiments is first covered as described in the next subsection.

5.1. Setup

Tables 1 and 2 outline the classification and regression datasets respectively, that will be used in our experiments showing details about the number of features and instances in each dataset. Two sources were used for the above datasets. The first is the UCI repository [40] and the second is MLData [41].
Table 1. Classification datasets.

| Name                      | Number of Features | Number of Instances |
|---------------------------|--------------------|---------------------|
| indian_liver_patient      | 10                 | 583                 |
| heart_disease             | 14                 | 303                 |
| HCV_Egy                   | 29                 | 1385                |
| mammogram                 | 6                  | 961                 |
| breast_cancer_wisconsin   | 11                 | 700                 |
| cryotherapy               | 8                  | 90                  |
| contraceptive_method      | 10                 | 1473                |
| pima_native_american_diabetes | 9              | 768                 |
| transfusion               | 5                  | 748                 |
| spect                     | 23                 | 267                 |

Table 2. Regression datasets.

| Name          | Number of Features | Number of Instances |
|---------------|--------------------|---------------------|
| slice_localization | 386               | 53,500              |
| CASP          | 9                  | 45,730              |
| risk_reclass  | 5                  | 1000                |
| mlt           | 6                  | 1200                |
| sj_ms2_fig4   | 3                  | 4000                |
| ocdata        | 6                  | 1200                |
| psa2b         | 6                  | 683                 |
| parkinsons    | 26                 | 5875                |
| fim           | 5                  | 59                  |
| health        | 5                  | 53                  |

The laptop configuration that was used in conducting the experiments is depicted in Table 3.

Table 3. Laptop configuration.

| Processor (CPU)          | Memory | System Type        |
|--------------------------|--------|--------------------|
| Intel(R) Core(TM) i7-5600U CPU @ 2.6GHz | 8.00 GB | 64-bit Operating System |

Table 4 outlines details about the metrics that will be reported in the experiments. The first two are common to both classification and regression datasets, the next three are for the classification datasets, and the last four are for the regression datasets.

Table 4. Metrics definition.

| Metric   | Definition                                      | Equation                                                                 |
|----------|------------------------------------------------|--------------------------------------------------------------------------|
| ITPI     | Inference Time per Instance (in microseconds) | Total Prediction Time/Number of Instances                                 |
| Pruning Level | Reduction Ratio Between eGAP and parentRF  | (1 – (Size(Pruned)/Size(Original))) × 100%                                 |
| Percentage Accuracy | Fraction of Correct Predictions | (Correct Predictions/Total Number of predictions) × 100% |
| AUC      | Area Under Curve                               | Probability(score(x+) > score(x−))                                      |
| F-Measure| Harmonic Mean of Precision and Recall          | (2 × Precision × Recall)/(Precision + Recall)                            |
| MAE      | Mean Absolute Error                            | 1/n \sum_{t=1}^{n} |e_t| |
| MSE      | Mean Squared Error                             | \frac{1}{n} \sum_{t=1}^{n} e_t^2 |
| RMSE     | Root Mean Squared Error                        | \sqrt{\frac{1}{n} \sum_{t=1}^{n} e_t^2} |
| R Squared| Coefficient of Determination                   | 1 – (Sum of Squares/Total Sum of Squares)                               |

5.2. Experiments and Results

The eGAP algorithm outlined in Algorithm 1 was implemented using the Python programming language utilising its machine learning library Scikit-learn to produce a variety of classification and regression metrics as was outlined in Table 4.
With reference to Algorithm 1, in our experiments, the following values were used for user settings. For both $S$ and $RD_{iterations}$, 1000 was used. Hence, the size of parentRF was 1000 trees, and 1000 RD iterations were applied to grow/shrink the working clusters during the evolution process. As for the number of clusters $k$, multiples of 5 in the range 5–50 were used.

To use the holdout testing method, which is the simplest type of cross validation, each dataset was divided into two sets: training and testing. A total of two thirds (66%) were reserved for training and the rest (34%) for testing. The selection of the two sets was done randomly using uniform distribution, where each instance has the same probability of being selected.

As demonstrated in the next subsection, the performance of eGAP will not be compared with parentRF only, but also with a random forest of identical size termed RF, where the trees in the forest are chosen at random from the parentRF, and with CLUB-DRF. The latter refers to the CLUB-DRF method used in [8,9] where clustering was the main technique used in the extreme pruning of random forests. Doing such a comparison between eGAP and CLUB-DRF can shed the light as to whether RD has the potential of improving the performance or not.

To compare the performance of eGAP with the parentRF, RF, and CLUB-DRF, the key performance indicator was used. This refers to percentage accuracy for classification and Mean Absolute Error (MAE) for regression. It is highlighted in boldface, in Tables 5 and 7 below, whenever eGAP performed at least as well as parentRF, RF, and CLUB-DRF. It is worth mentioning that though multiples of 5 clusters in the range 5–50 were generated as mentioned earlier, the best performing cluster from which eGAP was generated was the only one that was reported for each dataset. These are listed in the second column in the forthcoming Table 5 (for the classification datasets) and Table 7 (for the regression datasets).

5.2.1. Classification

The data in Table 5 compare the performance of eGAP with parentRF, RF, and CLUB-DRF on the classification datasets. As shown in this table, percentage accuracy, Area Under Curve (AUC), and F-Measure are reported.

| Dataset                          | Clusters | Accuracy | AUC  | F-Measure | Accuracy | AUC  | F-Measure |
|----------------------------------|----------|----------|------|-----------|----------|------|-----------|
| eGAP                             |          |          |      |           |          |      |           |
| indiana_liver_patient            | 50       | 91.97%   | 0.88 | 0.92      | 71.65%   | 0.59 | 0.72      |
| heart_disease                    | 45       | 92.76%   | 0.92 | 0.93      | 81.00%   | 0.80 | 0.81      |
| HCV_Egy                          | 5        | 74.77%   | NA   | 0.75      | 24.03%   | NA   | 0.24      |
| mammogram                       | 5        | 79.76%   | 0.80 | 0.80      | 78.32%   | 0.78 | 0.78      |
| breast_cancer_wisconsin         | 40       | 98.45%   | 0.98 | 0.98      | 93.87%   | 0.94 | 0.94      |
| cryotherapy                     | 10       | 98.86%   | 0.99 | 0.99      | 98.85%   | 0.99 | 0.99      |
| contraceptive_method            | 5        | 80.79%   | NA   | 0.81      | 52.44%   | NA   | 0.52      |
| pima_native_american_diabetes   | 5        | 90.91%   | 0.89 | 0.91      | 72.68%   | 0.67 | 0.73      |
| transfusion                     | 10       | 72.77%   | 0.58 | 0.73      | 72.03%   | 0.57 | 0.72      |
| spect                           | 5        | 74.88%   | 0.73 | 0.75      | 70.75%   | 0.69 | 0.71      |
| parentRF                        |          |          |      |           |          |      |           |
| indiana_liver_patient            | 50       | 91.97%   | 0.88 | 0.92      | 91.95%   | 0.88 | 0.92      |
| heart_disease                    | 45       | 92.76%   | 0.92 | 0.93      | 92.86%   | 0.92 | 0.93      |
| HCV_Egy                          | 5        | 74.77%   | NA   | 0.75      | 74.21%   | NA   | 0.74      |
| mammogram                       | 5        | 79.76%   | 0.80 | 0.80      | 77.93%   | 0.78 | 0.78      |
| breast_cancer_wisconsin         | 40       | 98.45%   | 0.98 | 0.98      | 98.40%   | 0.98 | 0.98      |
| cryotherapy                     | 10       | 98.86%   | 0.99 | 0.99      | 98.82%   | 0.99 | 0.99      |
| contraceptive_method            | 5        | 80.79%   | NA   | 0.81      | 80.75%   | NA   | 0.81      |
| pima_native_american_diabetes   | 5        | 90.91%   | 0.89 | 0.91      | 90.53%   | 0.89 | 0.91      |
| transfusion                     | 10       | 72.77%   | 0.58 | 0.73      | 72.14%   | 0.57 | 0.72      |
| spect                           | 5        | 74.88%   | 0.73 | 0.75      | 68.98%   | 0.67 | 0.69      |

RF
Table 5. Cont.

| Dataset                | Clusters | Accuracy | AUC  | F-Measure | Accuracy | AUC  | F-Measure |
|------------------------|----------|----------|------|-----------|----------|------|-----------|
| indian_liver_patient   | 50       | 91.97%   | 0.88 | 0.92      | 91.70%   | 0.88 | 0.92      |
| heart_disease          | 45       | 92.76%   | 0.92 | 0.93      | 92.26%   | 0.92 | 0.92      |
| HCV_Egy                | 5        | 74.77%   | NA   | 0.75      | 74.27%   | NA   | 0.74      |
| mammogram              | 5        | 79.76%   | 0.80 | 0.80      | 79.63%   | 0.80 | 0.80      |
| breast_cancer_wisconsin| 40       | 98.45%   | 0.98 | 0.98      | 98.28%   | 0.98 | 0.98      |
| cryotherapy            | 10       | 98.86%   | 0.99 | 0.99      | 98.06%   | 0.98 | 0.98      |
| contraceptive_method   | 5        | 80.79%   | NA   | 0.81      | 81.00%   | NA   | 0.81      |
| pima_native_american_diabetes | 5     | 90.91%   | 0.89 | 0.91      | 90.69%   | 0.89 | 0.91      |
| transfusion            | 10       | 72.77%   | 0.58 | 0.73      | 72.86%   | 0.58 | 0.73      |
| spect                  | 5        | 74.88%   | 0.73 | 0.75      | 74.95%   | 0.74 | 0.75      |

Table 5 demonstrates that eGAP has outperformed the parentRF, RF, and CLUB-DRF on almost all the datasets. As an interesting observation in the first column of this table, size 5 clusters appeared to be the mode, which means it was the best performing cluster size in the majority of the datasets.

For Inference Time per Instance (ITPI, rounded to the nearest integer) and pruning level (which only applies to the parentRF), the data in Table 6 compare the performance of eGAP with that of parentRF, RF, and CLUB-DRF. In this table, it is interesting to see that of the 10 datasets, eGAP, relative to the parentRF, achieved an extreme pruning level over 90% on six datasets and less than 90% on four datasets. Consequently, due to its smaller size compared with the parentRF, it achieved a faster ITPI on all the datasets as demonstrated in the third column.

Table 6. Inference Time per Instance (ITPI) and pruning level performance metrics of eGAP vs. parentRF, RF, and CLUB-DRF on classification datasets.

| Dataset                        | Clusters | ITPI    | Pruning Level | ITPI            |
|--------------------------------|----------|---------|---------------|-----------------|
|                                |          | eGAP    | parentRF, RF, CLUB-DRF |
| indiand Liver Patient          | 50       | 1387.05 | 69.30%        | 5734, 1437, 241 |
| heart Disease                  | 45       | 2956.75 | 77.60%        | 8770, 3356, 490 |
| HCV_Egy                        | 5        | 363.08  | 96.10%        | 4841, 378, 45   |
| mammogram                      | 5        | 137.62  | 96.60%        | 4052, 165, 18   |
| breast_cancer_wisconsin        | 40       | 1126.1  | 73.40%        | 3786, 1202, 151 |
| cryotherapy                    | 10       | 23,872.37 | 0.20%     | 29,228, 27,259, 323 |
| contraceptive_method           | 5        | 69.86   | 97.80%        | 3318, 74, 18    |
| pima Native American Diabetes  | 5        | 148.87  | 95.90%        | 4279, 187, 19   |
| transfusion                    | 10       | 211.77  | 94.60%        | 4000, 212, 39   |
| spect                          | 5        | 208.79  | 97.80%        | 9880, 198, 55   |

In the same table (Table 6 that is), though eGAP and RF had identical size, of the 10 datasets, eGAP interestingly achieved faster ITPI on eight datasets. This is likely due to eGAP having shorter trees. This may be attributed to the RD process, when better performing trees in the idle clusters joined the active clusters. Typically, shorter trees tended to overfit the data less, and hence had a better performance. Furthermore, in this table, when comparing the performance of eGAP with the CLUB-DRF method [8,9], which only utilised clustering, we saw that RD proved to be effective in improving the performance, since eGAP was able to outperform CLUB-DRF on seven datasets. For the three datasets where CLUB-DRF was superior over eGAP, the outperformance was a very small negligible fraction of less than 1%.

When comparing the ITPI between eGAP and CLUB-DRF, we saw that CLUB-DRF performed much better as it achieved much faster ITPI. This was expected and came as no surprise since in the CLUB-DRF method, only one representative was selected from each cluster yielding a much smaller ensemble than eGAP. At the conclusion of RD in the eGAP method, depending on how many trees were added to each working cluster, it was very likely that each working cluster would have many
trees (recall that after applying RD, the trees in the working clusters form the basis for the trees that eGAP will have as shown in the last for loop in Algorithm 1).

5.2.2. Regression

The data in Table 7 compare the performance of eGAP with the parentRF, RF, and CLUB-DRF on the regression datasets. As demonstrated in these tables, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R Squared are reported.

| Dataset Clusters | MAE | MSE | RMSE | R Squared |
|------------------|-----|-----|------|-----------|
| eGAP parentRF    |     |     |      |           |
| slice_localization | 0.45 | 4.89 | 2.21 | 0.99      |
| CASP             | 0.44 | 2.17 | 1.47 | 0.94      |
| risk_reclass     | 0.38 | 0.67 | 0.82 | 0.51      |
| mlit             | 13.01 | 236.53 | 15.38 | -0.19   |
| sj_ms2_fig4      | 0.92 | 1.39 | 1.18 | 0.35      |
| oedata           | 9.768 × 10⁻³ | 1.148 × 10⁻³ | 0.03 | 1.00 |
| psa2b            | 4.29 | 35.79 | 5.98 | 0.01 |
| parkinsons       | 3.07 × 10⁻⁴ | 1 × 10⁻⁴ | 7.13 × 10⁻⁴ | 1.00 |
| fim              | 129.72 | 66,340.13 | 257.56 | 0.50 |
| health           | 50.68 | 5106.97 | 71.44 | -0.31 |

| Dataset Clusters | MAE | MSE | RMSE | R Squared |
|------------------|-----|-----|------|-----------|
| eGAP parentRF    |     |     |      |           |
| slice_localization | 0.45 | 4.89 | 2.21 | 0.99      |
| CASP             | 0.44 | 2.17 | 1.47 | 0.94      |
| risk_reclass     | 0.38 | 0.67 | 0.82 | 0.51      |
| mlit             | 13.01 | 236.53 | 15.38 | -0.19   |
| sj_ms2_fig4      | 0.92 | 1.39 | 1.18 | 0.35      |
| oedata           | 9.768 × 10⁻³ | 1.148 × 10⁻³ | 0.03 | 1.00 |
| psa2b            | 4.29 | 35.79 | 5.98 | 0.01 |
| parkinsons       | 3.07 × 10⁻⁴ | 1 × 10⁻⁴ | 7.13 × 10⁻⁴ | 1.00 |
| fim              | 129.72 | 66,340.13 | 257.56 | 0.50 |
| health           | 50.68 | 5106.97 | 71.44 | -0.31 |

Table 7. Performance metrics of eGAP vs. parentRF, RF, and CLUB-DRF on regression datasets. (Figures in bold indicate the lowest MAE).

As demonstrated in Table 7, eGAP outperformed the parentRF, RF, and CLUB-DRF on almost all the datasets, with the mode being five clusters, as was the case with the classification datasets in Table 5.

Table 8 compares the ITPI and pruning level (which only applies to the parentRF) of eGAP relative to the parentRF, RF, and CLUB-DRF. In this table, note that of the 10 datasets, eGAP achieved an extreme pruning level over 90% on 7 datasets and less than 90% on three datasets. Consequently, due to its smaller size compared with the parentRF, it achieved a faster ITPI on all the datasets as demonstrated in the third column. When comparing eGAP and RF, though both have identical size, of the 10 datasets, eGAP achieved faster ITPI on eight datasets due to the likelihood that eGAP had shorter trees as stated before. Furthermore, as demonstrated in the table, since CLUB-DRF was much smaller in size than eGAP as discussed before (due to the fact only one representative was selected from each cluster yielding a much smaller ensemble), it was expected to have faster ITPI.
Table 8. ITPI and pruning level performance metrics of eGAP vs. parentRF, RF, and CLUB-DRF on regression datasets.

| Dataset          | Clusters | ITPI   | Pruning Level | ITPI   |
|------------------|----------|--------|---------------|--------|
| slice_localization | 5        | 134.48 | 97.80%        | 1889, 183, 10 |
| CASP             | 5        | 5.79   | 98.30%        | 345, 14, 2 |
| risk_reclass     | 40       | 178.89 | 57.80%        | 365, 179, 9 |
| mlt              | 25       | 4.89   | 97.50%        | 230, 18, 5 |
| sj_ms2_fig4      | 15       | 7.35   | 98.50%        | 74, 1, 1 |
| ocdata           | 50       | 278.74 | 21.00%        | 487, 323, 10 |
| psa2b            | 5        | 12.88  | 97.70%        | 391, 9, 9 |
| parkinsons       | 5        | 4.51   | 97.00%        | 251, 10, 1 |
| fim              | 5        | 428.61 | 88.00%        | 3381, 476, 143 |
| health           | 15       | 157.89 | 91.80%        | 3053, 158, 53 |

6. Discussion

In comparing the performance of eGAP with that of the parentRF for both the classification and regression datasets as demonstrated in Tables 5 and 7 respectively, we see that not only has eGAP outperformed parentRF on all the datasets, but even did so significantly for many datasets. Furthermore, as depicted in Tables 6 and 8, high pruning level was achieved on the majority of the datasets yielding an average pruning level of 71.58% for classification and 84.54% for regression. Consequently, faster ITPI was achieved, as demonstrated in these tables, since the number of trees in eGAP was significantly fewer than that of the parentRF. Likewise, eGAP performed consistently well relative to RF and CLUB-DRF as was demonstrated in Tables 5 and 7 respectively. Interestingly, it also achieved faster ITPI on the majority of the datasets as was demonstrated in these tables.

Since in the CLUB-DRF, clustering was also used, the performance gain achieved by eGAP over CLUB-DRF is likely attributed to RD. This is due to the fact that after completing the RD iterations, each working cluster eventually contains high performing trees as low performing trees are removed in each iteration. This is evident by the fact that eGAP was able, in most cases, to outperform the parentRF, RF, and CLUB-DRF.

As for the recommended cluster size, as seen in the experimental section, since five clusters had a recurring pattern, we recommend this cluster size for best results. Furthermore, favourable results, both in terms of the pruning level and accuracy, were achieved using five clusters. This makes eGAP a resource-efficient method suitable to develop smart healthcare applications.

eGAP is a novel method for ensemble pruning that is based on evolutionary game theory. The results reported in this article demonstrated its efficacy in pruning Random Forests as well as improving its predictive performance. It is worth noting that the method is generic, and can be used for pruning any ensemble model. Having used the smallest cluster size to determine the size of the initial active clusters in this work, other heuristics can be used to initialise the process. For example, initial active clusters can be of varying sizes. Such variation of settings may prove to be beneficial for larger ensemble models.

In the context of the Internet of Things (IoT), eGAP seems to fit well under the umbrella of healthcare smart cities. Relative to the traditional Random Forest, this is mainly attributed to its two-fold efficacy. The first is accuracy, and the second is speed due to its smaller size. These two traits are desirable in any smart application since they enable users to more efficiently complete a desired task or action. In the context of this paper, eGAP demonstrates its applicability to smart healthcare applications on both classification and regression tasks.

7. Conclusions and Future Work

In this research article, a new method has been introduced to improve the accuracy and speed of the traditional random forest’s classification and regression capabilities. The new method,
termed eGAP, produces a pruned version of the traditional random forest that is smaller in size than the original parent random forest from which it was derived and yet, performs at least as good as the parent random forest. To achieve this, eGAP combines two techniques that have been used before to improve random forests, namely, clustering and replicator dynamics. Clustering aims at producing clusters of similar trees based on their predictions on the training dataset. As was described in Section 4.3, after forming the idle and working clusters, replicator dynamics is then applied on each working cluster to grow/shrink the cluster by comparing its performance with the performance of all the working clusters. Due to the high pruning level achieved, and the improved performance over the traditional random forest, this makes eGAP a resource-efficient method that can form the basis for an ideal smart healthcare application.

Author Contributions: Conceptualization, M.M.G.; methodology, K.F. and M.M.G.; software, K.F.; writing—original draft preparation, K.F.; writing—review and editing, M.M.G. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Lytras, M.; Visvizi, A. Who uses smart city services and what to make of it: Toward interdisciplinary smart cities research. *Sustainability* 2018, 10, 1998. [CrossRef]
2. Chui, K.; Lytras, M.; Visvizi, A. Energy Sustainability in Smart Cities: Artificial intelligence, smart monitoring, and optimization of energy consumption. *Energies* 2018, 11, 2869. [CrossRef]
3. Lytras, M.; Visvizi, A.; Daniela, L.; Sarirete, A.; Ordonez De Pablos, P. Social networks research for sustainable smart education. *Sustainability* 2018, 10, 2974. [CrossRef]
4. Yan, J.; Liu, J.; Tseng, F.M. An evaluation system based on the self-organizing system framework of smart cities: A case study of smart transportation systems in China. *Technol. Forecast. Soc. Chang.* 2018, 153, 119371. [CrossRef]
5. Chui, K.T.; Alhalabi, W.; Pang, S.S.H.; Pablos, P.O.D.; Liu, R.W.; Zhao, M. Disease diagnosis in smart healthcare: Innovation, technologies and applications. *Sustainability* 2017, 9, 2309. [CrossRef]
6. Lytras, M.D.; Chui, K.T.; Visvizi, A. Data Analytics in Smart Healthcare: The Recent Developments and Beyond. *Appl. Sci.* 2019, 9, 2812. [CrossRef]
7. Fawagreh, K.; Gaber, M.M.; Elyan, E. Random forests: from early developments to recent advancements. *Syst. Sci. Control. Eng. Open Access J.* 2014, 2, 602–609. [CrossRef]
8. Fawagreh, K.; Gaber, M.M.; Elyan, E. Club-drf: A clustering approach to extreme pruning of random forests. In Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, 15–17 December 2015; pp. 59–73.
9. Fawagreh, K.; Gaber, M.M. Resource-efficient fast prediction in healthcare data analytics: A pruned Random Forest regression approach. *Computing* 2020, 1–12. [CrossRef]
10. Menger, V.; Scheepers, F.; Spruit, M. Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text. *Appl. Sci.* 2018, 8, 981. [CrossRef]
11. Ijaz, M.; Alfian, G.; Syyafrudin, M.; Rhee, J. Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. *Appl. Sci.* 2018, 8, 1325. [CrossRef]
12. Mun, S.; Park, J.; Dritschilo, A.; Collins, S.; Suy, S.; Choi, I.; Rho, M. The Prostate Clinical Outlook (PCO) Classifier Application for Predicting Biochemical Recurrences in Patients Treated by Stereotactic Body Radiation Therapy (SBRT). *Appl. Sci.* 2018, 8, 1620. [CrossRef]
13. Wang, J.; Wang, C.; Zhang, W. Data Analysis and Forecasting of Tuberculosis Prevalence Rates for Smart Healthcare Based on a Novel Combination Model. *Appl. Sci.* 2018, 8, 1693. [CrossRef]
14. Rady, E.H.A.; Anwar, A.S. Prediction of kidney disease stages using data mining algorithms. *Inform. Med. Unlocked* 2019, 15, 100178. [CrossRef]
15. Romeo, V.; Ricciardi, C.; Cuocolo, R.; Stanzione, A.; Verde, F.; Sarno, L.; Improta, G.; Mainenti, P.P.; D’Armiento, M.; Brunetti, A.; et al. Machine learning analysis of MRI-derived texture features to predict placenta accreta spectrum in patients with placenta previa. *Magn. Reson. Imaging* 2019, 64, 71–76. [CrossRef] [PubMed]

16. Manogaran, G.; Lopez, D. Health data analytics using scalable logistic regression with stochastic gradient descent. *Int. J. Adv. Intell. Paradig.* 2018, 10, 118–132. [CrossRef]

17. Nagarajan, V.; Kumar, D.V. An Optimized Sub Group Partition based Healthcare Data Mining in Big Data. *Int. J. Innov. Res. Sci. Technol.* 2018, 4, 79–85.

18. Kaur, P.; Kumar, R.; Kumar, M. A healthcare monitoring system using random forest and internet of things (IoT). *Multimed. Tools Appl.* 2019, 78, 19905–19916. [CrossRef]

19. Gaber, M.M.; Aneiba, A.; Basurra, S.; Batty, O.; Elmisery, A.M.; Kovalchuk, Y.; Rehman, M.H.U. Internet of Things and data mining: From applications to techniques and systems. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2019, 9, e1292. [CrossRef]

20. Mohindru, G.; Mondal, K.; Banka, H. Internet of Things and data analytics: A current review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2020, 10, e1341. [CrossRef]

21. Breiman, L. Random forests. *Mach. Learn.* 2001, 45, 5–32. [CrossRef]

22. Breiman, L. Bagging predictors. *Mach. Learn.* 1996, 24, 123–140. [CrossRef]

23. Ho, T.K. Random decision forests. In Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.

24. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* 1998, 20, 832–844.

25. Amit, Y.; Geman, D. Shape quantization and recognition with randomized trees. *Neural Comput.* 1997, 9, 1545–1588. [CrossRef]

26. Fawgreh, K.; Gaber, M.M.; Elyan, E. A replicator dynamics approach to collective feature engineering in random forests. In Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, 15–17 December 2015; pp. 25–41.

27. Fawagreh, K.; Gaber, M.M.; Elyan, E. Diversified random forests using random subspaces. In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Salamanca, Spain, 10–12 September 2014; pp. 85–92.

28. Kuncheva, L.I.; Whitaker, C.J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.* 2003, 51, 181–207. [CrossRef]

29. Brown, G.; Wyatt, J.; Harris, R.; Yao, X. Diversity creation methods: a survey and categorisation. *Inf. Fusion* 2005, 6, 5–20. [CrossRef]

30. Adeva, J.J.G.; Beresi, U.; Calvo, R. Accuracy and diversity in ensembles of text categorisers. *CLEI Electron. J.* 2005, 9, 1–12.

31. Tang, E.K.; Suganthan, P.N.; Yao, X. An analysis of diversity measures. *Mach. Learn.* 2006, 65, 247–271. [CrossRef]

32. Schuster, P.; Sigmund, K. Replicator dynamics. *J. Theor. Biol.* 1983, 100, 533–538. [CrossRef]

33. Taylor, P.D.; Jonker, L.B. Evolutionary stable strategies and game dynamics. *Math. Biosci.* 1978, 40, 145–156. [CrossRef]

34. Hofbauer, J.; Sigmund, K. Evolutionary game dynamics. *Bull. Am. Math. Soc.* 2003, 40, 479–519. [CrossRef]

35. Hilbe, C. Local replicator dynamics: A simple link between deterministic and stochastic models of evolutionary game theory. *Bull. Math. Biol.* 2011, 73, 2068–2087. [CrossRef] [PubMed]

36. Hauert, C. Replicator dynamics of reward & reputation in public goods games. *J. Theor. Biol.* 2010, 267, 22–28. [PubMed]

37. Nowak, M.A.; Sigmund, K. Evolutionary dynamics of biological games. *Science* 2004, 303, 793–799. [CrossRef] [PubMed]

38. Hauert, C.; De Monte, S.; Hofbauer, J.; Sigmund, K. Replicator dynamics for optional public goods games. *J. Theor. Biol.* 2002, 218, 187–194. [CrossRef]

39. Roca, C.P.; Cuesta, J.A.; Sánchez, A. Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics. *Phys. Life Rev.* 2009, 6, 208–249. [CrossRef]

40. Asuncion, A.; Newman, D. UCI Machine Learning Repository. 2007. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 26 November 2020).
41. Sharma, A. MLData. Available online: https://www.mldata.io/ (accessed on 26 November 2020).

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).