Journal of Medical Systems (2019) 43: 96
https://doi.org/10.1007/s10916-019-1200-1

Improving the Accuracy of Feature Selection in Big Data Mining Using Accelerated Flower Pollination (AFP) Algorithm

K. Venkatasalam[1] · P. Rajendran[2] · M. Thangavel[3]

Received: 1 November 2018 / Accepted: 11 February 2019 / Published online: 9 March 2019

Abstract
In recent times, the main problem associated with big data analytics is its high dimensional data over the search space. Such data gathers continuously in search space making traditional algorithms infeasible for data mining in real time environment. Hence, feature selection is an important method to lighten the load during processing while inducing a model for mining. However, mining over such high dimensional data leads to formulation of optimal feature subset, which grows exponentially and leads to intractable computational demand. In this paper, a novel lightweight mechanism is used as a feature selection method, which solves the after effects arising with optimal feature selection. The feature selection in big data mining is done using accelerated flower pollination (AFP) algorithm. This method improves the accuracy of feature selection with reduced processing time. The proposed method is tested under larger set of data with high dimensionality to test the performance of proposed method.

Keywords Accelerated flower pollination (AFP) algorithm · Data mining · Feature selection · Big data mining

Introduction

In recent times, the researchers found the big data has undergone certain level of degradation due to three issues. It includes velocity, variety and volume issues. The first one leads to handling of huge data [1–7] at accelerating high speed. The second issue leads to poor processing of data and poor integration of data from different sources and it is formatted contrarily. Finally, the third issue leads to processing, storage and analysis on archives and computational challenges. Depending on the outlook of the issues arising in big data stream, the conventional data mining techniques operated on full batch learning and it runs poor due to its demand in analytic efficiency. This includes two conventional algorithms, namely, regression tree algorithm and rough-set discrimination. The poor efficiency is due to loading of full dataset and then partitioning it using divide-and-conquer strategy. This data collection technique bloats the big data into bigger data, as soon as the newer data arrives. It makes the system to re-run the entire model and it has to be built newly for the arrival of new data [8].

On contrarily, there are several new breed algorithm [9–11] or data mining models, which diminishes the three issues. The new breed algorithm limits the stemming of huge data volume or data collection at high speed. Instead, it uses prediction and classification with bottom-up approach. This approach helps to update itself without reading past data and it handles infinite data streams by analyzing the memory and mining the data streams. This is considered as a killer method by the researches to solve the issues in big data stream. In addition, there are other algorithms like novel ensemble map-reduce paradigm [12], data parallelism map-reduce paradigm [13], Density-Peaked Clustering Analysis algorithm [14], i^2 map reduce incremental framework [15], association rule mining [19] with Apriori algorithm and probabilistic graphical model [16], multi-objective genetic programming [18] and Distributional instance learning

This article is part of the Topical Collection on *Mobile & Wireless Health*

✉ K. Venkatasalam
venkatasalamk@mahendra.info

[1] Department of Computer Science & Engineering, Mahendra Engineering College, Namakkal 637503, India

[2] Department of Computer Science, Knowledge Institute of Technology, Kakapalayam, India

[3] Department of Electronics and Communication Engineering, Knowledge Institute of Technology, Kakapalayam, India

[17–20]. It is found that these algorithm can provides solutions related to the issues of big data in future years [21]. In both batch and stream based algorithms of big data mining, classification is been adopted widely to support the decisions of big data model. In classifier application, the classes are predicted from the unseen samples and feature selection selects the significant features by discarding redundant and irrelevant features to improve the accuracy. Further, it aims to improve the speed and training time of classifier to solve the problems associated with feature selection. In case of supervised learning, a classifier induces the attributes based on records and labels from all data. Conversely, reports [22] say that conventional data mining models are limited due to three design constraints, which includes fixed size of feature set, minimal features using redundancy principle and customized feature selection, which works for one particular classifier. However, finding the relevant feature subset requires exhaustive computing, but this is impractical in big data streams. Since, the data collected will be of infinite amount with high speed.

To resolve this, this paper inspects the effectiveness of meta-heuristic light weighted feature selection using Accelerated Flower Pollination Algorithm (AFPA), which is derived from Flower Pollination Algorithm (FPA) [23, 24]. This aims at identifying the perfect blend of feature selection and classification algorithm for data mining in big data streams. The processing big data streams are collected from UCI Machine Learning Repository archives for testing the machine learning algorithms. This paper focusses on comparing the performance of batch learning classification algorithms to attain top accuracy with shortest processing time. The main challenge lies in finding appropriate algorithm for data mining. The challenges lies on infinite data feed, high speed continuous data delivery. Therefore, the processing is expected to be responsive and real-time. This ensures that the deployed classification algorithm is accurate and light weighted with quick update with faster data arrival. Another complication lies in non-linear relations between the target classes and feature values in temporal data stream. Also, the poor availability of straight forward model leads to poor mapping of data attributes into relevant classes. This affects the design of mining algorithm that should read and forget the stream of data without retaining anything but the statistics for providing reason for long-term relation between the target classes and feature values [25].

Considering all such assumptions associated with computational challenge, the proposed data streaming model compares itself with popular mining algorithm to tests its efficacy. The evaluation offers insights for the newer researches to design a mining application in big data stream. The organization of paper is presented as follows: Section 2 provides the feature selection model by proposed accelerated flower pollination algorithm. Section 3 incorporates the proposed algorithm into data streaming model. Section 4 discusses the evaluation method and classification models for testing the proposed design. Section 5 discusses the efficacy of proposed method. Finally, section 7 concludes the paper.

Feature selection by AFPA

The proposed method uses modern feature selection algorithm to choose the optimal subset from the large space. The proposed algorithm is called as Accelerated Feature Selection with Flower pollination Algorithm or Accelerated Flower pollination Algorithm (AFPA). The proposed algorithm is been designed as a wrapper based model for feature selection. It helps to maintain the rate of accuracy of classifiers been constructed from feature subset. This attains highest fitness and probable output is considered based on optimal feature subset. The architecture of the workflow of the AFPA model is shown in Fig. 1. This model refines the rate of accuracy by selecting better feature after the pre-selection of random feature subset at initial stage. This tends to increase the classification accuracy of proposed model in a stochastic manner. This helps to empower the classification model for optimal feature selection with higher and faster convergence rate [26].

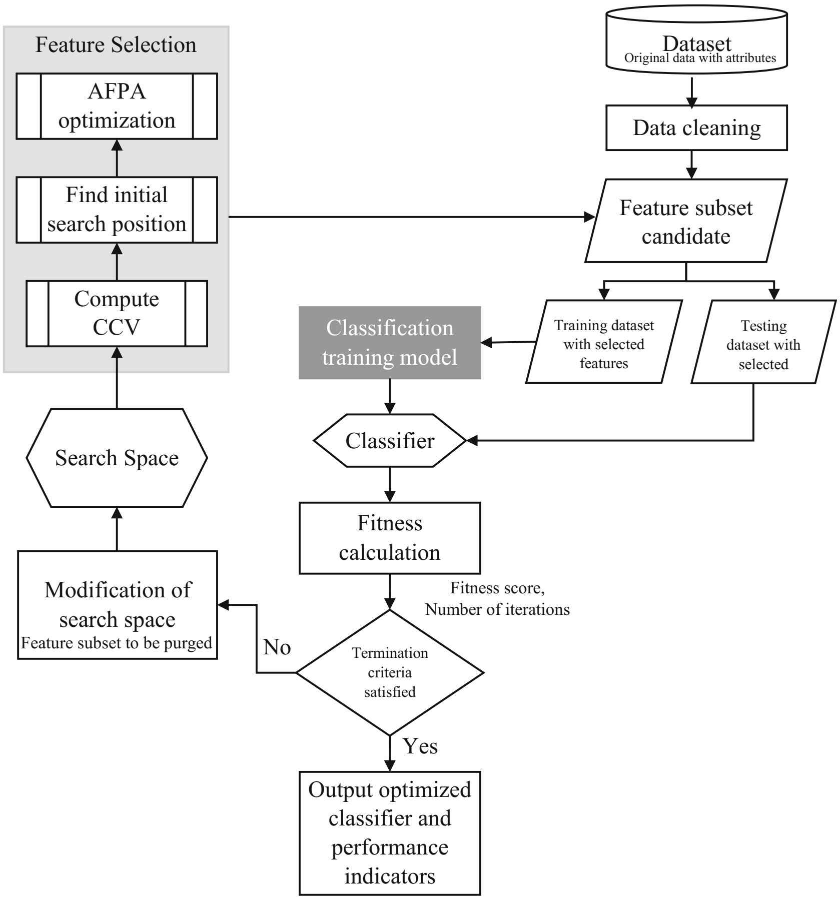
The fitness evaluation is chosen as wrapped classifier, which makes most use of the candidate feature subset with optimization function in stochastic manner. However, such brute force testing on all subsets takes longer time with increased high computation costs due to increasing data features. In such regards, the proposed method uses stochastic search strategy. This method avoids individual feature subset testing and promotes multi agent searching with parallel computation to find optimal feature subsets. Further, the search process is reduced using speed up step in flight behavior called accelerated FPA [27].

Accelerated FPA

The flower pollination algorithm (FPA) works on the principle of pollination through agents or bees on flowering plants or flowers. The aim of FPA is the survival of optimal reproduction of plants (features) or in fact, it is considered as optimizing the plant species. It is ruled by basic rules, which is given bellow:

a. As a part of global pollination process, the cross and biotic pollination is chosen with pollinators, which carries the pollens using the principle of Levy Flights.

Fig. 1 Architecture of proposed system


Feature Selection
AFPA optimization
Find initial search position
Compute CCV
Search Space
Modification of search space
Feature subset to be purged
Dataset
Original data with attributes
Data cleaning
Feature subset candidate
Training dataset with selected features
Testing dataset with selected
Classification training model
Classifier
Fitness calculation
Fitness score, Number of iterations
No
Termination criteria satisfied
Yes
Output optimized classifier and performance indicators

b. As a part of local pollination process, the self and abiotic pollination is considered.

c. The reproduction probability is chosen based on Flower constancy, which is directly related to the similarity between two different flowers or features involved in the dataset [8, 28, 29].

d. Switching probability (p) to control the global and local pollination, where $p \in [0, 1]$. This takes into account the factors like wind and physical proximity, which makes the local pollination to be an important fraction p through entire pollination activity.

During the process of global pollination (rule a and c), pollinators or insects carry the flower pollens or the relevant features. The Levy flights distribution makes the pollens to carry over longer distances and the global pollination is represented as,

$$x_i^{(t+1)} = x_i^t + \alpha L(\lambda)\left(g*-x_i^t\right) \quad (1)$$

Where,

$$L(\lambda) = \frac{\lambda \Gamma(\lambda) \cdot \sin(\lambda)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad s > 0$$

Where,

x_i^t pollenor feature i during the iteration t and
s step size,
α scaling factor for controlling the step size,
$g*$ best solution among all solutions during the iteration t,
$L(\lambda)$ Levy flight step size proportional to pollination strength and
$\Gamma(\lambda)$ gamma function and λ varies between $1 \leq \lambda \leq 2$ and in the present study, it is chosen to be 1.5.

Similarly, the process of local pollination (rule b) is represented as,

$$x_i^{(t+1)} = x_i^t + \varepsilon\left(x_j^t - x_k^t\right) \quad (2)$$

Where,

x_j^t pollen from the flower j of same plant and
x_k^t pollen from the flowers k of same plant.

Finally, to mimic the global and local pollination, the proposed study uses the rule d or the switching probability to choose between the global or local pollination. The pseudo code the FPA algorithm is given below,

Algorithm:

```
1. Fitness Function minimum or maximum $f(x)$, where $x = (x_1, x_2..., x_n)$;
2. Initialize the $n$ pollen gametes or flowers population through a random solution;
3. Identify the solution $g*$(best pollen gamete) in initial population;
4. Set the switch probability $p \in [0, 1]$;
5. while ($t$<maximum generation) do
    a. for$i = 1 : n$ (total flowers population) do
        i. if rand <$p$(switching probability), then
            1. Induce a d-dimensional step vector $L$with Levy flight distribution;
            2. Undergo Global pollination using Eq.(1);
        ii. else
            1. Induce uniform distribution in [0,1];
            2. Choose randomly $j$ and $k$flowers among all solutions;
            3. Undergo local pollination via Eq.(3);
        iii. end if
    b. Evaluate new solutions;
    c. if new solutions> previous solution, update the new solution in flower population; else use previous solution end
    d. end for
6. Estimate the present best solution $g*$ ;
7. end while
```

To speed up the process of convergence, the location of flowers are updated in a single step. Hence, for global pollination, the updated location is given by,

$$x_i^{(t+1)} = (1-\beta)x_i^t + \alpha L(\lambda)\left(\beta g^* - x_i^t\right) \quad (3)$$

Table 1 Performance evaluation between various classifiers

Classifier or Feature Selection	Accuracy	Precision	Recall	F-measure	Model building time (s)	Pre-processing time (s)	% Selected Features
Hyper-pipe (HP)							
PSO	80.253448	0.82096	0.80365	0.79855	0	0	100
APSO	74.913224	0.84433	0.76185	0.75877	0	0	47
FPA	81.026588	0.85082	0.80678	0.79059	0	2.03	73
AFPA	81.924583	0.85966	0.81079	0.80365	0	2.01	65
Naive Bayes (NB)							
PSO	68.351584	0.75883	0.68635	0.70478	0.0912	0	100
APSO	72.011458	0.76999	0.72105	0.73229	0.0101	0	47
FPA	82.014826	0.81983	0.82101	0.81589	0.0098	9.52	31
AFPA	81.829918	0.81171	0.81384	0.80675	0	9.52	25
Bayes Net (BN)							
PSO	84.588982	0.86282	0.84955	0.85567	0.37	0	100
APSO	84.018515	0.85567	0.85464	0.85665	0.0301	0	47
FPA	87.878464	0.87609	0.87813	0.87609	0.414	17.08	51
AFPA	88.986676	0.88729	0.88035	0.88832	0.402	11.73	39
Decision Tree (DT)							
PSO	89.014821	0.90361	0.91259	0.90259	0.203	0	100
APSO	89.485689	0.89463	0.90656	0.90463	0.1101	0	47
FPA	91.248483	0.91789	0.91893	0.91892	0.1101	37.41	47
AFPA	92.151247	0.91994	0.92197	0.92995	0.061	20.3	27
Random Forest (RF)							
PSO	93.025646	0.93032	0.93033	0.93319	0.2052	0	100
APSO	94.825722	0.95664	0.95767	0.95359	0.0816	0	47
FPA	94.236824	0.94237	0.94339	0.93727	0.0713	35.27	43
AFPA	95.128972	0.96379	0.96482	0.96277	0.1022	32.06	27
Support Vector Machine (SVM)							
PSO	77.194902	0.69267	0.77815	0.68327	2.0004	0	100
APSO	77.84202	0.69795	0.78631	0.70267	0.8777	0	47
FPA	77.952681	0.76387	0.78733	0.70675	0.3906	362.32	29
AFPA	79.826122	0.77307	0.79957	0.73023	0.3215	305.61	27
Neural Network (NN)							
PSO	91.912574	0.92607	0.92921	0.92595	25.0621	0	100
APSO	90.924856	0.91789	0.91893	0.91677	8.5004	0	47
FPA	91.098564	0.91892	0.92197	0.91897	14.0402	5455.73	71
AFPA	92.652142	0.92903	0.93319	0.92013	11.7536	5001.84	65

Similarly, the local pollination is updated using,

$$x_i^{(t+1)} = (1-\beta)x_i^t + \varepsilon\left(\beta x_j^t - \beta x_k^t\right) \tag{4}$$

Since, these equations increases the speed of convergence and accelerates the process of pollination, the FPA algorithm can now be called as accelerated FPA or AFPA.

The proposed mining model uses effective feature selection using coefficient of variation for finding the initial position of AFPA. The coefficient of variation is based on variance, which finds the feature subset for balancing the classifier in an optimal manner between the problem of features getting over-fitted and generalization. The coefficient of variation is founded based on the belief that an attribute has varying range of values in a training dataset, such that it characterizes the prediction model. The coefficient of variation varies between - ∞ and + ∞, which is represented as a real number and this forms a standard deviation relative to mean value of set of numbers. This helps to compare the variability and it informs the degree of variation w.r.t observation size and units of observation is not directly related with coefficient of variation.

However, the variation coefficient is not similar for all feature and it is not relative to the measuring unit. Hence, the information about the variation of data among all features is obtained through coefficient of variation. Here, if the expected value is mean, then expected variability is coefficient of variation related to mean [30–32]. Thus when multiple heterogeneous data is present in the data set, the expected variability can be used for multiple measurement on same dataset. Even if the data is measured on different scale, the coefficient of variation between two attributes for feature selection is compared directly. On other hand, the standard deviation is obtained through absolute relative variation. This is known as dispersion measure, which compares differences between the variables of varying units and the higher variation coefficient variable is considered dispersed more than lower coefficient variable.

Consider a training dataset (X) with n vector instances and m features or attributes. Consider an instance or sample (x_1, x_2,...,a_m) with m-dimensional tuple, where a ∈ [1,2,…,m] for a vector x_a. This is segregated into various subgroups of varying classes with total prediction target classes (c∈[1,2,...,C]). Hence, $x_a \in \left\{x_a^1, x_a^2, \ldots, x_a^c\right\}$,

$$v_a = \sum_{c=1}^{C} \frac{\sqrt{\sum_{n=1}^{N} \left(x_n^c - \overline{x}_a^c\right)^2 / n}}{\overline{x}_a^c} \tag{5}$$

Where,

$\overline{x}_a^c$ mean of a^{th} feature belonging to class c and

v_a sum of coefficient of variation of a^{th} feature belonging to class c.

The coefficient of variation finds the threshold value after estimating the coefficient variable to select the features and estimates the retained features. This concept is called as Bia-Variance dilemma, where the prediction performance of a learning classifier is found by decomposing the supervised classifiers error rate into variance and bias terms.

Consider a target function: $x = x +$ 휀 over a fixed training size set (D) is used for expressing the expected squared error (X,T) into three sum components:

$$\sum_D \left[\int_x \int_t (h(x) - t)^2 p(t|x) p(x) dt dx\right] = \sigma^2 + b^2 + \text{var}^2 \tag{6}$$

$$\sigma^2 = e_a \tag{7}$$

$$b^2 = \int \left(\sum_D [h(x)] - g(x)\right)^2 p(x) dx \tag{8}$$

$$\overline{h}(x) = \sum_D [h(x)] \tag{9}$$

$$\text{var} = \int \sum_D \left[\left(h(x) - \overline{h}(x)\right)^2\right] p(x) dx \tag{10}$$

where,

b bias and var – variance,

The main aim is to reduce the loss by decomposing into variance, sum of bias and noise term (constant). Further, there exist a trade-off between variance and bias with relative rigid model (low variance and high bias) forms under fit and very flexible model (high variance and low bias) forms overfit. Hence, to attain optimal equilibrium between variance and bias, k-means clustering is used. This partitions the point into clusters, where one cluster can be retained and other one is removed. Each data point is assigned with a cluster membership function. The clustering find the cluster position μ_i,i = 1,2,...,k and distance between the data point and centroid of cluster is reduced using,

$$Fitness = \min_c \sum_{i=1}^{2} \sum_{x \in c_i} d(x, \mu_i) \tag{11}$$

$$\arg\min_c \sum_{i=1}^{2} \sum_{x \in c_i} \|x - \mu_i\|^2 \tag{12}$$

where c_i- set of points of cluster i. The clustering uses (x,μ_i) = ||x-μ_i||^2, which is the square of Euclidean distance.

Consider a dataset X with estimation function f(x) = $\overrightarrow{a}\overrightarrow{x}$, where $\overrightarrow{a}\overrightarrow{x} = a_0x_0 + a_1x_1 +, \ldots..$ It is seen that adding parameters into AFPA as features increases the complexity of the model and hence variance increases with falling bias. Here, k-means clustering is deployed to form two groups (g_1 and g_2) from the given dataset based on the coefficient of variation. Further, the complexity of model increases as the variance and bias values of data points of different clusters are not the same. Hence, the complexity is reducing by the selection of relevant attributes through variance separation. The error of two k-means group is given below,

$$g_1 = b_2\uparrow + \text{var}\downarrow + \sigma^2 \tag{13}$$

$$g_2 = b_2\downarrow + \text{var}\uparrow + \sigma^2 \tag{14}$$

The groups in Eq.(13) and Eq.(14) represents the combination of bias and variance for selecting the optimal subset of features. The up arrow represents the increasing values of bias or variance and down arrow represents the decreasing values of bias or variance.

Evaluation method

The experiment uses various classifiers of traditional batch learning to evaluate the performance of proposed light

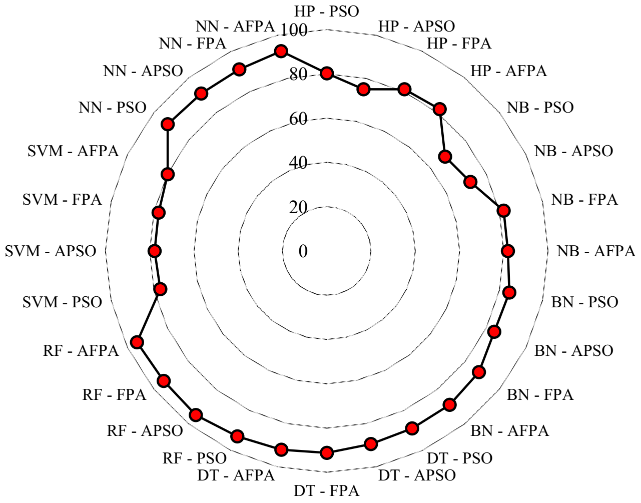


Fig. 2 Accuracy between proposed and conventional models with various classifiers using radar chart

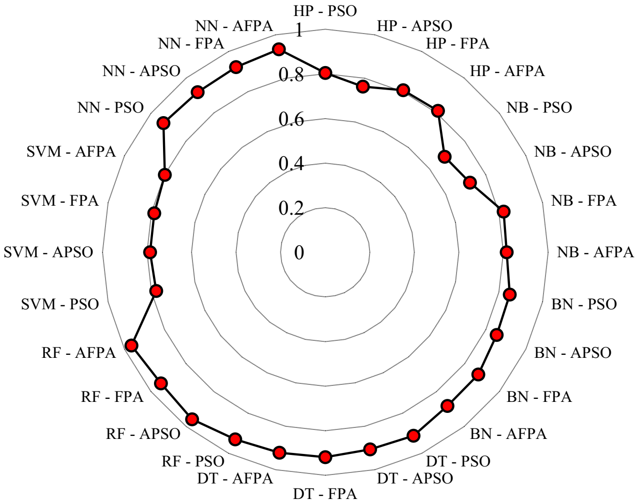


Fig. 4 Recall between proposed and conventional models with various classifiers using radar chart

weight classification algorithm with existing algorithms like FPA (flower Pollination Algorithm), PSO (Particle Swarm Optimization) and APSO (Accelerated PSO). The different classifiers considered for evaluation include: Naïve Bayes, Bayes Net, neural network, support vector machine, random forest, Hyperpipe and decision tree. The performance of proposed and conventional method is tested in terms of accuracy, precision, recall, F-measure, percentage of features selected, total model building time of classifier and total pre-processing time of classifier [4, 33, 34]. The results are shows in Table 1.

The experiment between the proposed model and classifier is conducted on Dell T7610 PC computing platform with 128GB RAM on Intel Xeon Processor E5. The environment chosen is Java Development Kit and the classifiers are implemented with the help of MOA platform. For the experimental purpose, the parameters are set as default ones. To test the algorithms in an effective way, a 10-fold cross-validation is utilized to attain unbiased estimate of the proposed and conventional classification model. Here, the dataset is fragmented into ten equal subsets and unseen data is used for validate the performance. The classifier models are build using the similar algorithms, which are built around ten rounds by sparing one subset to train the given model.

Big data stream classification

The big data streams subjected for evaluation uses four preprocessing algorithms for proper feature selection. The first one includes PSO, which is set as ground truth value, since it uses original form of datasets from UCI archives. The APSO is correlation based type for selecting the features from the archives. The FPA uses firefly strategy to select the features

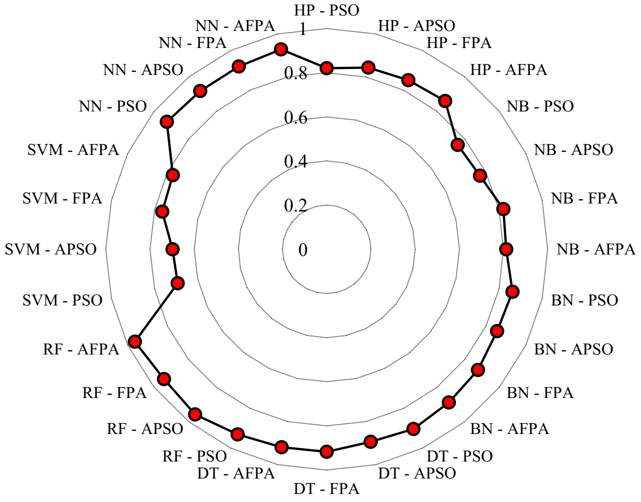


Fig. 3 Precision between proposed and conventional models with various classifiers using radar chart

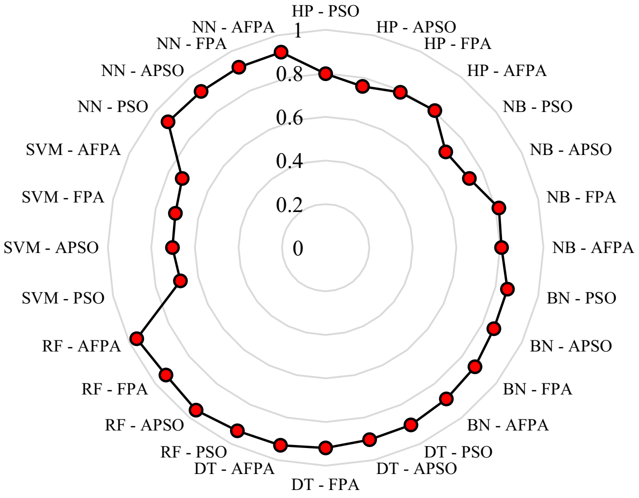


Fig. 5 F-measure between proposed and conventional models with various classifiers using radar chart

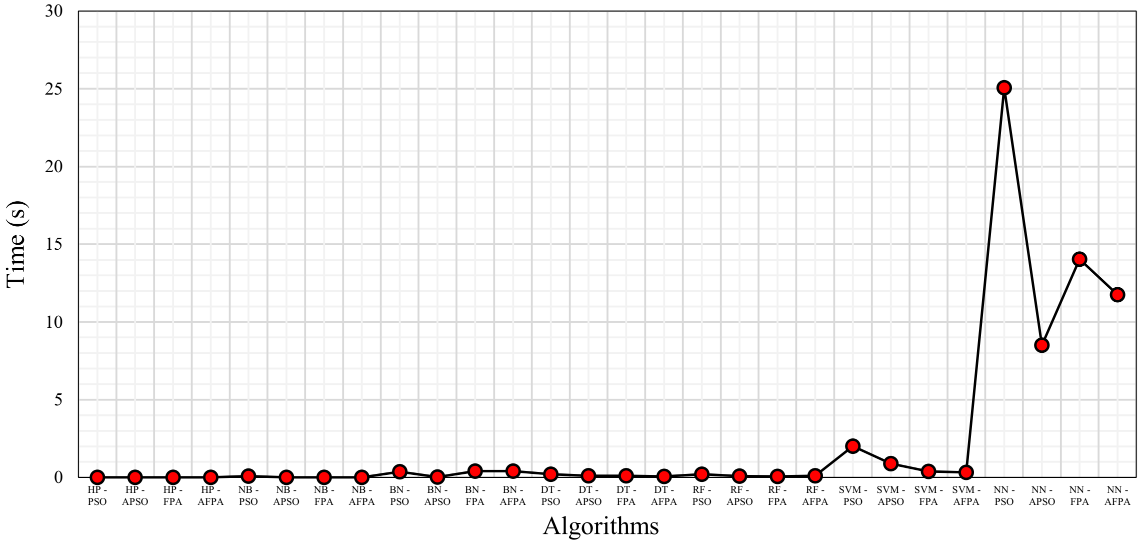


Fig. 6 Model Building time between proposed and conventional models with various classifiers

and finally APSO is used to select the features using correlation with firefly characteristics. The third and fourth method is otherwise called as feature selection PSO and APSO or FS-PSO and FS-APSO, respectively.

Results and discussion

The Fig. 2 shows the accuracy of between proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of accuracy. The results between conventional and proposed streaming model shows that proposed system attains higher accuracy. Here, Random forest classifier attains higher accuracy than neural network, decision tree, Bayes Net, Hyper-pipe, Naïve Bayes and SVM. This proves the random forest classifier can be used for future researches to model the big data streaming model.

The Fig. 3 shows the precision of between proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of precision. The results between conventional and proposed streaming model shows that proposed system attains higher precision. However, in terms of Naïve Bayes classifier, the FPA algorithm attains higher precision rate. Here, Random forest classifier attains higher precision than neural network, decision tree, Bayes Net, Hyper-pipe, Naïve Bayes and SVM.

The Fig. 4 shows the recall of between proposed AFPA and other conventional data streaming models like FPA,

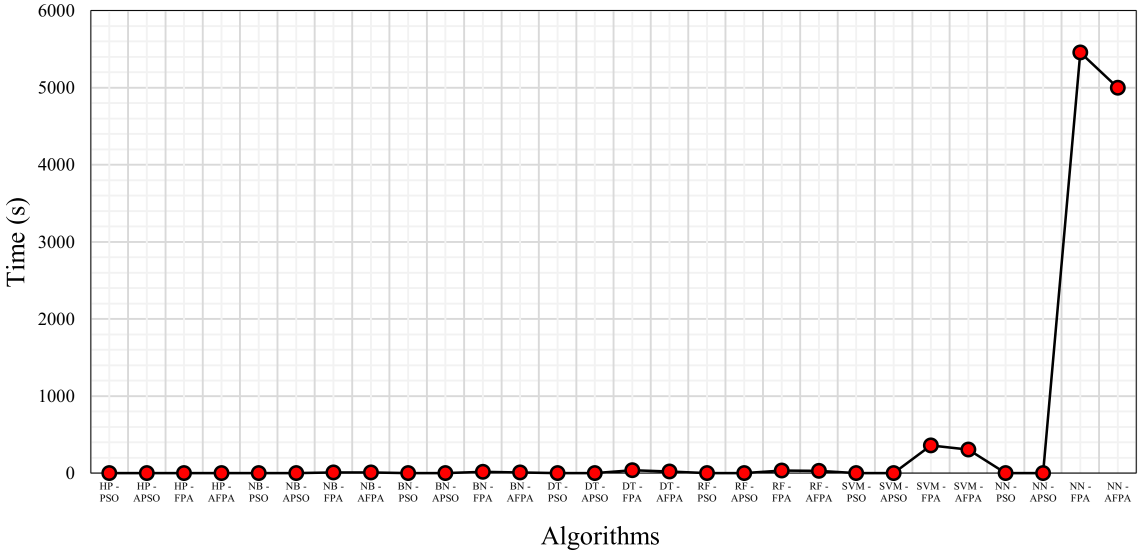


Fig. 7 Processing time between proposed and conventional models with various classifiers

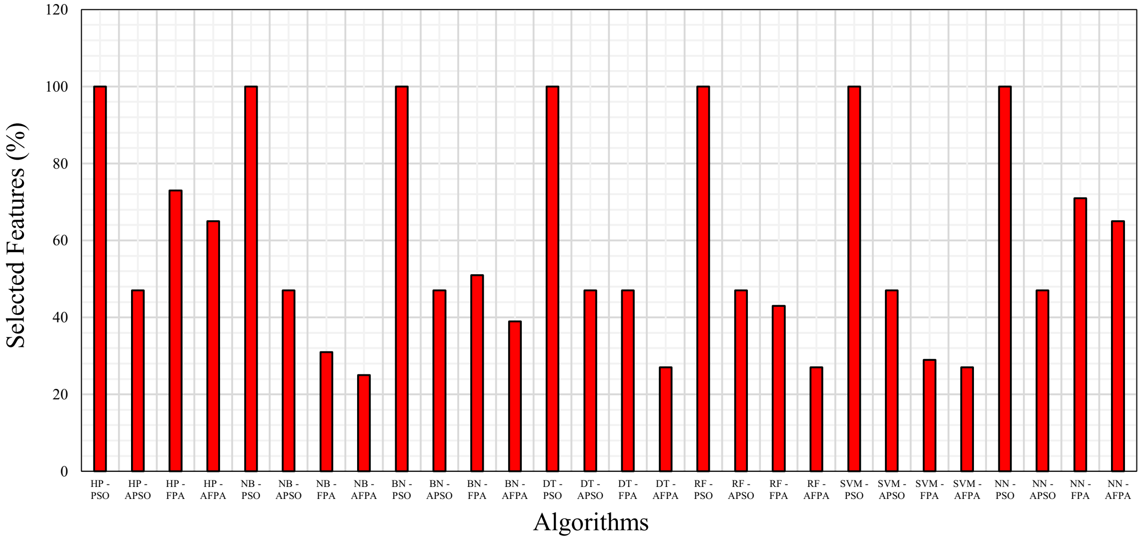


Fig. 8 Percentage of selected features between proposed and conventional models with various classifiers

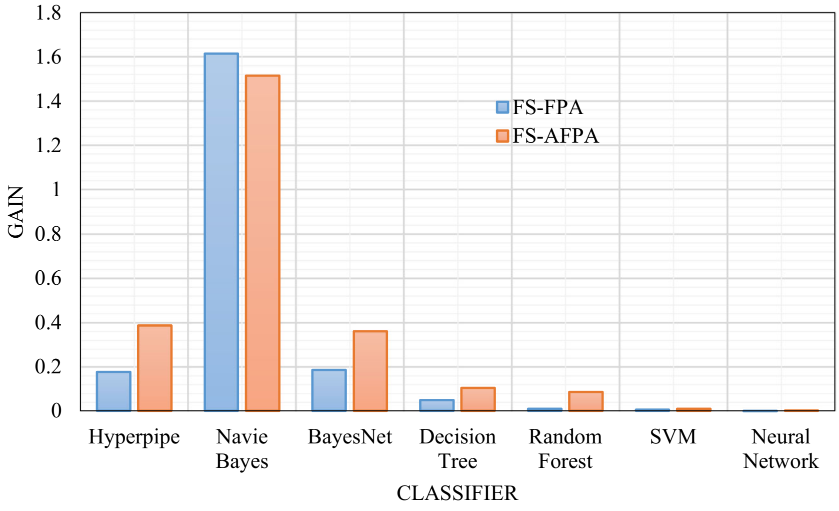
PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of recall. The results between conventional and proposed streaming model shows that proposed system attains higher recall rate. However, in terms of Naïve Bayes classifier, the FPA algorithm attains higher recall rate. Here, Random forest classifier attains higher recall than neural network, decision tree, Bayes Net, Naïve Bayes Hyper-pipe and SVM.

The Fig. 5 shows the F-measure of between proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of F-measure. The results between conventional and proposed streaming model shows that proposed system attains higher F-measure. However, in terms of Naïve Bayes classifier, the FPA algorithm attains higher F-measure. Here, Random forest classifier attains higher F-measure than decision tree, neural network, Bayes Net, Naïve Bayes Hyper-pipe and SVM.

The Fig. 6 shows the building time of model between the proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of model building time. The results between conventional and proposed streaming model shows that proposed system attains lesser model building time. However, in terms of Random Forest classifier and neural network classifier, the PSO and APSO algorithm, respectively attains lesser model building time. Here, Naïve Bayes and Hyper-pipe classifier attains lesser building time than decision tree, Random forest, SVM, Bayes Net, and neural network.

The Fig. 7 shows the pre-processing time of between proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons

Fig. 9 Performance Gain between proposed and conventional models with various classifiers



between the proposed and conventional data streaming models with different classifier in terms of pre-processing time. The results between conventional and proposed streaming model shows that proposed system attains lesser pre-processing time. However, the results shows that PSO and APSO algorithm takes zero time to compute pre-process the required inputs. This shows that the proposed AFPA and FPA model, still needs to be retuned in terms of its operating speed associated with pre-processing operations. Here, Hyper-pipe and Naïve Bayes classifier attains lesser pre-processing time than Bayes Net, decision tree, Random forest, SVM, and neural network.

The Fig. 8 shows the percentage of selected features of between proposed AFPA and other conventional data streaming models like FPA, PSO and APSO. The Table 1 shows the comparisons between the proposed and conventional data streaming models with different classifier in terms of percentage of selected features. The percentage of feature selection in proposed method is very less and this can be interpreted as efficient collection of relevant data samples related to given query. This is not true in case of PSO and APSO algorithm, where it selects entire instances, which choosing the relevant ones. The poor pre-processing operation leads to such selection of features in conventional system than the proposed system. Here, Naïve Bayes classifier attains lesser percentage of feature selection than decision tree, Random forest, SVM, Bayes Net, Hyper-pipe and neural network.

To measure the fairness of proposed system, a gain indicator is used as a performance increasing factor, which considers the accuracy during the process of feature selection. Hence the incremental accuracy is estimated as follows,

Gain or incremental accuracy = percentage of accuracy during feature selection – percentage of original accuracy at preprocessing in terms of milli-seconds

Ideally, the algorithms are combined in such a way that it offers higher gain i.e. yielding higher accuracy in short preprocessing time. The Fig. 9 shows the comparison of classifiers in terms of gain indicator. From the Table 2 or Fig. 9, it is seen that Naïve Bayes classifier attains higher gain than other classifiers. The neural network classifier performs with poor gain in big data streaming model and hence it performs poor with fresh data arrival in big data streams. Also, the other classifiers like Hyper-pipe, Bayes Net, Decision Tree, Random Forest and SVM offers least gain improvement, which is lesser than Naïve Bayes algorithm. Finally, it could be concluded that the proposed lightweight accelerated FPA algorithm offers higher gain than flower pollination algorithm. The Naïve Bayes algorithm attain higher performance gain due to less selection of features, in comparison with other systems.

Table 2 Performance Gain between proposed and conventional models with various classifiers

Classifier	FS-FPA	FS-AFPA
Hyper-pipe	0.175675	0.385175
Naive Bayes	1.614038333	1.514860795
Bayes Net	0.183847954	0.358581818
Decision Tree	0.048638444	0.1031754
Random Forest	0.008184481	0.084807503
SVM	0.004679534	0.008606064
Neural Network	−0.000156751	8.00808E-05
Average	0.290700999	0.350755237

Conclusion

In this paper, the high dimensionality of data in big data streaming is addressed, since incomplete data provides higher computational demand in data mining applications. Since the big data grows with fresh data, a light weight mechanism is required to monitor it dynamically in a larger scale. The algorithm offers higher robustness, reduced latency and high accuracy. The proposed classification model classifies well the big data stream over five different dataset domains, having wide features. The comparison of various classifiers with proposed light weight AFPA classification model is proved to be fruitful in data stream mining. The results shows that proposed method with various classifiers attains higher gain at preprocessing for the selection of features. This paper provides various insights on different classifiers, who are intended to design a new data mining model for selecting the features in big data analytics. The proposed lightweight AFPA feature selection approach fits to be better for arriving data streams in real world applications and it helps to meet the big data demands in service computing.

Compliance with Ethical Standards

Conflict of Interest The author's has no conflict of interest in submitting the manuscript to this journal.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Fan, W., and Bifet, A., Mining big data: Current status, and forecast to the future. ACM sIGKDD Explor. Newslet. 14(2):1–5, 2013.
2. Fong, S., Yang, X. S., Deb, S. Swarm search for feature selection in classification. In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on (902–909). IEEE. 2013.

3. Sundarasekar, R., Thanjaivadivel, M., Manogaran, G., Kumar, P. M., Varatharajan, R., Chilamkurti, N., and Hsu, C. H., Internet of things with maximal overlap discrete wavelet transform for remote health monitoring of abnormal ECG signals. J. Med. Syst. 42(11): 228, 2018.
4. Kumar, P. M., Lokesh, S., Varatharajan, R., Babu, G. C., and Parthasarathy, P., Cloud and IoT based disease prediction and diagnosis system for healthcare using fuzzy neural classifier. Futur. Gener. Comput. Syst. 86:527–534, 2018.
5. Kumar, P. M., Devi, U., Manogaran, G., Sundarasekar, R., Chilamkurti, N., and Varatharajan, R., Ant colony optimization algorithm with internet of vehicles for intelligent traffic control system. Comput. Netw. 144:154–162, 2018.
6. Vijayakumar, V., Priyan, M. K., Ushadevi, G., Varatharajan, R., Manogaran, G., and Tarare, P. V., E-health cloud security using timing enabled proxy re-encryption. Mob. Netw. Appl.:1–12, 2018.
7. Parthasarathy, P., and Vivekanandan, S., Investigation on uric acid biosensor model for enzyme layer thickness for the application of arthritis disease diagnosis. Health Inform. Sci. Syst. 6(1):–6, 2018.
8. Mathan, K., Kumar, P. M., Panchatcharam, P., Manogaran, G., and Varadharajan, R., A novel Gini index decision tree data mining method with neural network classifiers for prediction of heart disease. Des. Autom. Embed. Syst.:1–18, 2018.
9. Priya, S., Varatharajan, R., Manogaran, G., Sundarasekar, R., and Kumar, P. M., Paillier homomorphic cryptosystem with poker shuffling transformation based water marking method for the secured transmission of digital medical images. Pers. Ubiquit. Comput.:1–11, 2018.
10. Varatharajan, R., Preethi, A. P., Manogaran, G., Kumar, P. M., and Sundarasekar, R., Stealthy attack detection in multi-channel multi-radio wireless networks. Multimed. Tools Appl.:1–24, 2018.
11. Manogaran, G., Shakeel, P. M., Hassanein, A. S., Priyan, M. K., and Gokulnath, C., Machine-learning approach based gamma distribution for Brian abnormalities detection and data sample imbalance analysis. IEEE Access. 2018.
12. Fong, S., Liang, J., and Wong, R., Ghanavati, M. A novel feature selection by clustering coefficients of variations. In digital information management (ICDIM), 2014 ninth international conference on (205-213). IEEE., 2014.
13. Parthasarathy, P., and Vivekanandan, S., A numerical modelling of an amperometric-enzymatic based uric acid biosensor for GOUT arthritis diseases. Inform. Med. Unlocked., 2018.
14. Parthasarathy, P., and Vivekanandan, S., Urate crystal deposition, prevention and various diagnosis techniques of GOUT arthritis disease: A comprehensive review. Health Inform. Sci. Syst. 6(1):19, 2018.
15. Bouckaert, R. R., Bayesian network classifiers in weka for version 3-5-7. Artif. Intel. Tools 11(3):369–387, 2008.
16. Parthasarathy, P. Synthesis and UV detection characteristics of TiO2 thin film prepared through sol gel route. In IOP Conference Series: Materials Science and Engineering (Vol. 360, No. 1, p. 012056). IOP Publishing. 2018.
17. Basha, A. A., Vivekanandan, S., and Parthasarathy, P., Evolution of blood pressure control identification in lieu of post-surgery diabetic patients: A review. Health Inform. Sci. Syst. 6(1):17, 2018.
18. Varadharajan, R., Priyan, M. K., Panchatcharam, P., Vivekanandan, S., and Gunasekaran, M., A new approach for prediction of lung carcinoma using back propogation neural network with decision tree classifiers. J. Ambient. Intell. Humaniz. Comput.:1–12, 2018.
19. Zhou, Z., Chen, S., and Chen, Z., FANNC: A fast adaptive neural network classifier. Knowl. Inf. Syst. 2(1):115–129, 2000.
20. Huang, C. L., Chen, M. C., and Wang, C. J., Credit scoring with a data mining approach based on support vector machines. Expert Syst. Appl. 33(4):847–856, 2007.
21. Verikas, A., Gelzinis, A., and Bacauskiene, M., Mining data with random forests: A survey and results of new tests. Pattern Recogn. 44(2):330–349, 2011.
22. Parthasarathy, P., and Vivekanandan, S., A typical IoT architecture-based regular monitoring of arthritis disease using time wrapping algorithm. Int. J. Comput. Appl.:1–11, 2018.
23. Parthasarathy, P., and Vivekanandan, S., A comprehensive review on thin film-based nano-biosensor for uric acid determination: Arthritis diagnosis. World Rev. Sci. Technol. Sustain. Dev. 14(1): 52–71, 2018.
24. Lior, R. Data mining with decision trees: theory and applications (Vol. 81). World scientific. 2014.
25. Kranjc, J., Orač, R., Podpečan, V., Lavrač, N., and Robnik-Šikonja, M., ClowdFlows: Online workflows for distributed big data mining. Futur. Gener. Comput. Syst. 68:38–58, 2017.
26. Tsai, C. F., Lin, W. C., and Ke, S. W., Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. J. Syst. Softw. 122:83–92, 2016.
27. Chen, J., Li, K., Rong, H., Bilal, K., Yang, N., and Li, K., A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. Inf. Sci., 2018.
28. Zhang, Y., Chen, S., Wang, Q., and Yu, G., I $^ 2$ mapreduce: Incremental mapreduce for mining evolving big data. IEEE Trans. Knowl. Data Eng. 27(7):1906–1919, 2015.
29. Sheng, G., Hou, H., Jiang, X., and Chen, Y., A novel association rule mining method of big data for power transformers state parameters based on probabilistic graph model. IEEE Trans. Smart Grid. 9(2):695–702, 2016.
30. Wu, X., Zhu, X., Wu, G. Q., and Ding, W., Data mining with big data. IEEE Trans. Knowl. Data Eng. 26(1):97–107, 2014.
31. Gandomi, A. H., Sajedi, S., Kiani, B., and Huang, Q., Genetic programming for experimental big data mining: A case study on concrete creep formulation. Autom. Constr. 70:89–97, 2016.
32. Afzali, G. A., and Mohammadi, S., Privacy preserving big data mining: Association rule hiding using fuzzy logic approach. IET Inf. Secur., 2017.
33. Lokesh, S., Kumar, P. M., Devi, M. R., Parthasarathy, P., and Gokulnath, C., An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map. Neural Comput. & Applic.:1–11, 2018.
34. Somasekhar, G., Karthikeyan, K. The novel big data algorithm for distributional instance learning. Ain Shams Engineering Journal, In press corrected proof. 2017.