An Improved Distributed Fuzzy Associative Classifier for Big Data Using Butterfly Optimization Algorithm based Artificial Bee Colony Algorithm

T.B. Saranya Preetha* & Dr.S. Jayasankari**

*Research Scholar, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam, Tamilnadu, INDIA.
E-Mail: saranyapreethaa{at}gmail{dot}com
**Associate Professor, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam, Tamilnadu, INDIA.
E-Mail: jayasankaris{at}pkrarts{dot}org

***Abstract*—**Association rule mining is one of the primary tasks in data mining, which is helpful in knowing the intriguing associations among the elements in the item sets of a massive database. Association Rule Mining is a prominent data mining approach. Researchers have evaluated several factors of the approach; however, very less focus is paid to tackle with the reliability of the rules external to the dataset using which the generation of these rules is done. Aproiri constitutes the common algorithm of association rule mining that helps frequent item sets' generation. Apriori utilizes minimal support threshold to get frequent items. In this work, an algorithm, formed by Butterfly Optimization Algorithm based Artificial Bee Colony Algorithm is presented which helps in choosing the association rules with Associative Classifiers (ACs). Rather than the ABC's onlooker bee stage, the random walk process of BOA is utilized to improve the exploration. Butterfly Optimization Algorithm Based Artificial Bee Colony Algorithm (BOAABC) is used on the rules that the apriori algorithm generates, to choose the association rules. The validation process is carried out on datasets obtained from UCI database which reveal the performance of the proposed research work and the proposed technique is efficient in the choice of association rules rather than the available algorithms. In this work, it is confirmed that the rules created in the proposed research work are easy and understandable.

***Keywords*—**Associative Classifiers (ACs); Artificial Bee Colony (ABC); Butterfly Optimization Algorithm (BOA); Frequent Item Sets.

***Abbreviations*—**Associative Classifiers (ACs); Artificial Bee Colony (ABC); Butterfly Optimization Algorithm (BOA); Butterfly Optimization Algorithm Based Artificial Bee Colony Algorithm (BOAABC).

I. INTRODUCTION

SEVERAL association rule mining algorithm operate on the basis of the supposition that the items existing in the dataset belong to the same type with identical frequencies. Therefore, the algorithms utilize level wise support thresholds for mining. If the item sets have diverse frequency and variable significance, the level wise support thresholds are inappropriate to find the frequent correlations. Every item in a level shows diverse features and routines [Altaf et al., 1]. In order to yield reduced support and high confidence association rules, it is essential to define the support thresholds for every item. This technique, depending on Apriori, is an extension of the association rule model by permitting the user to define different support thresholds so that the changing characteristics and/or frequencies of items can be shown. Utilizing these defined multiple support thresholds, particular itemsets can be pruned during the step of frequent item setgeneration, when eliminating several rules during the of rules generation phase [Abdel-Basset et al., 2].

In this research work, Artificial bee colony algorithm with Butterfly Optimization Algorithm based association rule selection approach has been proposed. The remaining section of this work is structured as given. Section 2 presents the literature reviewand Association rule mining is studied. Section 3 studies about the proposed technique. Section 4 provides the results of the eexperiments and the setup for performance comparison outcomes. Lastly, the conclusion of the research work is discussed in Section 5.

II. LITERATURE REVIEW

Qureshi et al., [3] studied about a WalkBackABC model, which helps in the optimization of ARM by improving the exploration region and helps in the optimization of the association rules and its individual comparison is performed using apriori, FP growth and ABC algorithms and the test outcomes shows the rules formed, which are straightforward and understandable.

Segatori et al., [4] carried out an elaborate experiment and a comprehensive analysis on six massive datasets with over 11 000 000 instances and the results are studied. In this, it can be noticed that, even though the result of the accuracies are reasonable, the complexity is much lesser compared to one of the non fuzzy classifiers.

Lin et al., [5] developed Multiple Fuzzy Frequent Itemset (MFFI)-Miner algorithm for mining MFFIs. Two effective pruning mechanisms are also developed to limit the search space belonging to the enumeration tree depending on the fuzzy-list setup. Experiments revealed that the proposed techniques offer superior mining performance compared to other available models.

Zoraghchian et al., [6] introduced the butterfly optimization algorithm (BOA), which offers a satisfying accuracy and speed in finding a solution to optimization problems, is considered for ARM. The evaluation of the proposed technique reveals its superior performance in terms of accuracy and execution time.

Telikani et al., [7] suggested a novel rule hiding algorithm that depends on a binary Artificial Bee Colony (ABC) technique known as Improved Binary ABC (IBABC) which is merged with proposed rule hiding algorithm. Also, the efficiency of IBABC is validated applying the incapacitated facility location problem and 0–1 knapsack problem.

Nguyen et al., [8] suggested a new algorithm referred as ETARM (Efficient Top-k Association Rule Miner), which is efficient in discovering the entire set of top-k association rules. An elaborate test analysis on six benchmark datasets confirms that the performance of the proposed technique is better than the benchmark Top K Rules algorithm both with regard to runtime and memory utilization.

III. PROPOSED METHODOLOGY

The proposed system developed an Improved Distributed Fuzzy Associative Classifier (IDFAC) with Butterfly Optimization Algorithm Based Artificial Bee Colony Algorithm for big data. The Apriori algorithm is merged with optimal rule selection process using the extracted frequent Fuzzy Classification Association Rules (FCARs). Here, an algorithm developed through Butterfly Optimization Algorithm Based Artificial Bee Colony Algorithm (BOAABC) isused for choosing the optimal rules. At last, the rules selected with the application of certain mechanisms, which rely on support, confidence, and distributed training set coverage. Figure 1 shows the flow diagram of the proposed technical work.

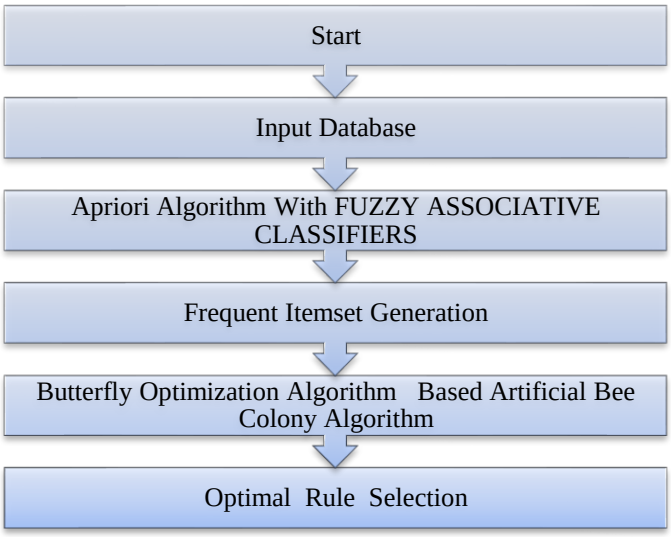

Figure 1: Architecture Diagram

3.1. Association Rule Mining

Learning of Association rules is helpful in discovering the associations between features in massive databases. An association rule, A=> B, will specify a group of transactions, a value assigned to itemset A decides the values assigned to itemset B under the criteria in which minimum support and confidence are satisfied. Support and Confidence for Itemset A and B are given by expressions:

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence (A}\longrightarrow\text{B)} = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

Apriori algorithm belonged to the group of the primary algorithms introduced for frequent itemset mining [Wang et al., 9; Zheng et al., 10]. This algorithm utilizes two steps, which are join and prune so that the search spacegets reduced. It is an iterative mechanism usedto find the most frequent itemsets. Apriori states that the probability that item I is infrequent is when:

P(I) < minimum support threshold, then I is rare.

P (I+A) < minimum support threshold, then I+A is rare, where A also falls under itemset.

In case, an itemset setis of the value lesser compared to minimum support then each one of its supersets will again be lesser than min support, and therefore can be missed

The mth Fuzzy Classification Association Rule (FCAR) formed out of the rule base RB = {FCAR1, . . . , FCARM} is therefore defined as,

$$\text{FCAR}_m: \text{FAnt}_m \rightarrow C_{l_m} \text{with} \text{RW}_m \quad (1)$$

Where the consequent C_{l_m} indicates the class label chosen for the rule and the antecedent FAnt_m is given to be the conjunction.

$$\text{FCAR}_m : \text{IF } X_1 \text{ is } A_{1,l_{m,1}} \text{ AND}\ldots. X_F \text{ is } A_{F,m} \quad (2)$$

Where the FS Af, m specifies variable Xf indicates either to an FS $A_{f,jf,m}$ in the group for attribute Xf, or to the entire universe

Uf (in this final scenario, the membership degree of any value is 1, and the respective term is true at all times). The number of terms in (2), which are true sometimes only is called as rule length, and is given by gm. Noticeably, in (1) the rule is paired with a weight RWm, to define its relative significance each and every time utilized along with every other rule.

The level to which a particular rule FCARm is a match to an example on = (xn, yn) is defined through the strength of activation (or matching degree with the input), computed as,

$$\mathrm{Wm(xn)} = \prod_{f=1}^{F} \mu_{f,m}\left(x_{f,n}\right) \quad (3)$$

Where, $\mu_{f,m}(x)$indicates the membership function corresponding to the FS $A_{f,m}$.

During association rule analysis, support and confidence constitute the generic factors to decide the intensity of an association rule, with regard to the Training Set TS. As with a common $\mathrm{FCAR_m}$ they can be defined as,

$$\text{Fuzzy Support}(\mathrm{FAnt_m} \rightarrow C_{l_m}) = \frac{\sum_{x_n \in TS_{l_m}} w_m(x_n)}{N} \quad (4)$$

$$\text{fuzzyConf}(\mathrm{FAnt_m} \rightarrow C_{l_m}) = \frac{\sum_{x_n \in TS_{l_m}} w_m(x_n)}{\sum_{x_n \in TS} w_{FAnt_m}(x_n)} \quad (5)$$

3.2. Artificial Bee Colony Algorithm with Butterfly Optimization Algorithm

In this FCAR technical work, artificial bee colony is utilized for rule selection. Artificial bee colony (ABC) algorithm is a population-based stochastic optimization. The ABC algorithm simulates the food foraging nature of the actual honey bees [Neelima et al., 11]. In ABC algorithm, the food source for bees is known as solutions. ABC consists of three kinds of bees, which are the employed, the onlooker and the scout. In ABC colony, the count of employed and onlooker bees are equal. Employed bee move to the food sources and return to the hive and share the information with onlooker bee by a dance on the dance floor. Onlooker bee observethe dances and selects the food sources based on the dance movements. The employed bee whose food sources have been neglected goes on to be a scout and begins looking for anotherfood source.

In the first stage, the ABC initializes the features in the data, which is considered as the input in place of the food source. Every solution Xi (i=1, 2,…, SN) indicates a D-dimensional vector where D refers to the number of parameters that have to be optimized [Ameta, 12]. The population of the locations (search process of the employed, onlooker and scout) is continued till the Maximum Cycle Number (MCN), C=1,2,…MCN is attained.

An employed bee makes a change on the location applying Eq. (6). In this proposed technical work, the quantity of nectar is taken as classification accuracy. An employed bee makes a change on the source location that it has memorized and finds another feature location. In the case that the classification accuracy of the new one is more than the earlier one, the bee saves the new bees location in its memory and discards the old one from its memory. Else it retains the position of the old one in its memory.

$$v_{ij} = x_{ij} + \emptyset_{ij}\left(x_{ij} - x_{kj}\right) \quad (6)$$

Where, k∈{1,2,…,SN} and j∈{1,2,…,D} are randomly picked indexes; k is arbitrarily set and must be different from i, and Øij is a randomly formed number in the range [−1,1].

Once each one of the employed bees finish the search process, they exchange the information of their food source location with the onlooker bees by doing waggle dances. An onlooker bee assesses the classification accuracy and the features selected with a probability, pi, associated with its classification accuracy adopting Eq. (7):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (7)$$

Where, fit_i refers to the fitness value of the solution i and SN indicates the number of features in the data. The employed bee makes a change in the location and verifies the classification accuracy of the candidate source. In case, the accuracy is more compared to the earlier one, the onlooker bee keeps the new location in its memory and forgets the old one.

The feature whose accuracy is discarded by the bees is substituted with the fresh features using the scouts by Eq. (8) if no further improvement in the position is possible. The parameter "limit" constitutes the control parameter to decide the discarding of the features within the predefined number of cycles.

$$x_i^j = x_{min}^j + \mathrm{rand}(0,1)\left(x_{max}^j - x_{min}^j\right) \quad (8)$$

In order to deal with the disadvantages of artificial bee colony (ABC) algorithm, which involves a slow convergence during the rule selection process and early convergence, in the proposed mechanism, the minimum and maximum value of random variable is formed using Butterfly Optimization Algorithm (BOA) [Faris et al., 13; Dubey, 14], and the rule selection is improved. In order to conceptualize the above definitions in terms of a search algorithm, the above mentioned attributes of butterflies are defined as below: 1. All the butterflies are expected to discharge a kind of scent that facilitates the butterflies to get attracted to one another. 2. All the butterflies will traverse either in random or towards the best butterfly discharging increasing scent. 3. The strength of stimulus of a butterfly is influenced or decided in terms of the space of the objective function. Three stages in BOA are as follows: (1) Initialization stage, (2) Iteration stage and (3) Final stage. Depending on every run of BOA, at first, the initialization step is run, and next search process is carried by iteration and later in the final stage, the algorithm gets stopped once the best solution is got.

During every run of BOA, at first during the initialization stage, the algorithm specifies the objective function along with its solution space. The parameter values utilized in BOA also get initialized. Once the values are set, the algorithm continues generating the first level population of butterflies for the optimization process. Since the overall number of butterflies stays unmodified during the BOA simulation, a predefined size memory is assigned for storing their information [Lambert & Perumal, 15]. The locations of butterflies are formed in ad-hoc in the search space, and their scent and fitness values are computed and saved. This completes the initialization stage and then begins the iteration stage of the algorithm, which

carries out the search using the synthesized butterflies generated.

The second step of the algorithm, which is the iteration phase, involves several iterations the algorithm runs. During every iteration, all the butterflies present in solution space traverse to new locations and later their fitness values are assessed. In the first step, the algorithm computes the fitness values of every butterfly on various locations in the solution space. Later, these butterflies will produce the scent at their locations applying Eq. (2). The algorithm consists of two stages, which includes global search stage and local search stage. In global search stage, the butterfly moves a step towards the fittest butterfly/solution g∗and it can be found employing Eq. (9)

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (9)$$

Where x_i^t refers to the solution vector xi assigned for ith butterfly during iteration number t. In this, g∗ indicates the current best solution got out of each of the solutions in the present iteration. firefers to the scent of ith butterfly and r stands for an arbitrary number in the range [0, 1]. Local search stage can be formulated as

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i \quad (10)$$

$$\vec{F}(t+1) = xi \quad (11)$$

Where x_j^t and x_k^tspecifies the jth and kth butterflies taken from the solution space. If x_j^t and x_k^t ar from the same swarm and r stands for any arbitrary number in the range [0, 1] then it is a local random walk. Looking for food and partners to mate by butterflies can happen both locally and globally. $\vec{F}(t+1)$ indicates the final feature, which would be combined with ABC. Taking the physical vicinity and several other factors such as rain, wind, etc., into consideration, searching for food can impose a considerable ratio p in the overall mating partner or food look out behaviour of butterflies. Therefore, a change probability p is utilized in BOA to change between the general search globally to a strong search locally. At last, the update feature is computed as below, The expression below is used for computing the feature for selection of rule.

$$x_i^j = x_{min}^j + \vec{F}(t+1)\left(x_{max}^j - x_{min}^j\right) \quad (12)$$

Algorithm 2: Optimal Rules Selection

Input: Number of rules
Output: Optimal rules
Decide the number of rulesx_i, i = 1…N
Decide rules position
Assess accuracy of the rules
Set cycle to 1
Repeat
FOR each employed bee
Generate new solutions vi by applying (6)
Compute the accuracy
Use the greedy selection process
Compute the probability pi for the solution $x_i$applying (7)
FOR each onlooker bee
Choose a solution xi based on pi
Generate new solutions vi
Compute the accuracy
Use the greedy selection process
If a discarded solution exists for scout bees then
Substitute it with a new solution using expression(12)
Memorize the best solution (optimal rules) attained till now
cycle = cycle + 1
Until cycle = MCN (Maximum cycle number)
end

IV. EXPERIMENTAL RESULTS

The implementation of the proposed research is done in MATLAB simulation environment. The performance evaluation of the proposed approach is carried out on datasets, whose data are obtained from UCI database. The comparison of the proposed technique is done with FP-ABC and FR-GWOABC algorithms. The evaluation is carried out on the basis of the expression,

This section studies the results acquired from EDFAC-FFP, confirming its remarkable accuracy compared to those attained using the contemporary benchmark techniques like DFAC-FFP, and MRAC. The parameters utilized for the classification comparison include precision, F-measure, accuracy, recall and error rate.

$$\text{Precision} = \frac{TP}{TP+FP} \text{ x } 100 \quad (13)$$

$$\text{Recall} = \frac{TP}{TP+FN} \text{ x } 100 \quad (14)$$

$$\text{F measure} = 2\,\frac{\text{precision} \,.\, \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FN + FP + TN)} * 100 \quad (16)$$

$$\text{Error rate} = 100 - \text{Accuracy} \quad (17)$$

Here, TP, TN, FP and FN signifies True Positive, True Negative, False Positive, and False Negative.

Table 1: Comparison Analysis of Classification

Methods/ Metrics	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)	Error rate(%)
MRAC	57.16946	59.038462	58.088931	74.266667	25.733333
FP-ABC	79.942056	71.589429	70.754578	78.092	21.908
FR-BOAABC	83.636364	87	86.666667	86.666667	13.333333

"Table 1" shows the comparison analysis of different rule mining approaches in terms of precision, recall, F-measure, accuracy and error rate/

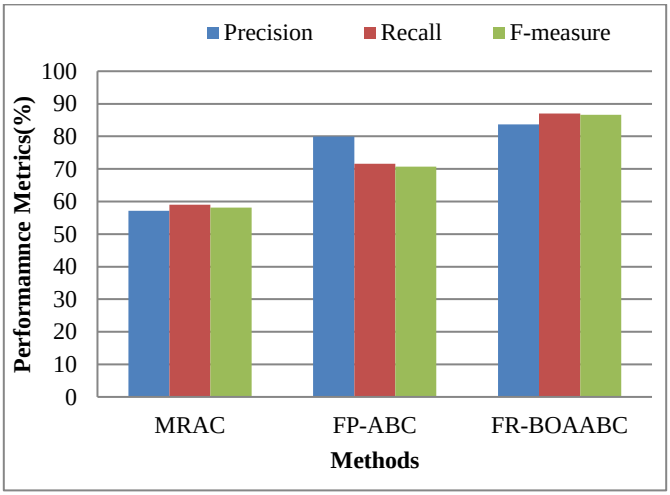

Figure 2: Precision, Recall and F-Measure Results Comparison

The comprehensibility measure is realized in the proposed approach so that the rules are made clear and perceivable. "Figure.2" illustrates the comprehensibility with respect to Precision, Recall and F-measure comparison between the proposed and available algorithms. "Figure 2" depicts the comparison analysis performed between different rule mining approaches in terms of precision. The precision for proposed FR-BOAABC is achieved at 83.6364 %, recall for proposed FR-BOAABC is attained at 87 and F-measure for proposed FR-BOAABC is attained at 86.666 which is comparatively more than the available MRAC and FP-ABC.

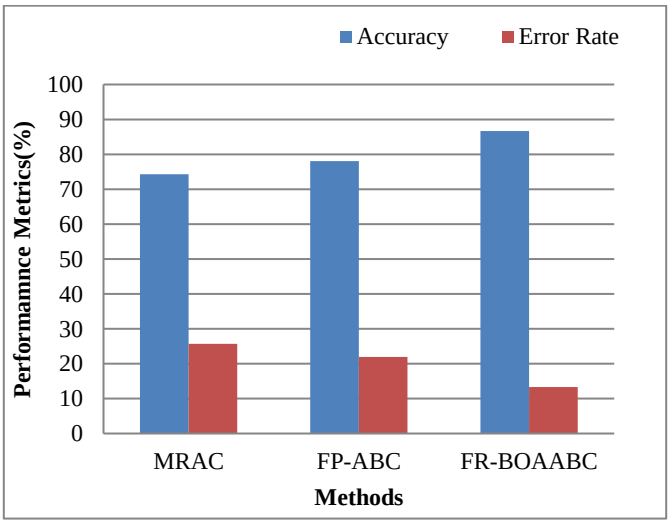

Figure 3: Accuracy and Error Rate Results Comparison

It is evident from "Figure 3" that the rules produced by the proposed approach of accuracy and error rate are optimal in comparison with available algorithms and the rules produced using the proposed technique are clear. The Accuracy for proposed FR-BOAABC is attained at 86.666 %, Error Rate for proposed FR-BOAABC is achieved at 13.33 which is comparatively more than the available MRAC and FP-ABC. Therefore, the performance achieved using the proposed technique works is excellent in comparison with available algorithms.

V. CONCLUSION

In this proposed technical work, a modified form of Distributed Fuzzy Associative Classifier (DFAC) with Butterfly Optimization Algorithm based Artificial Bee Colony (BOAABC) algorithm is designed for big data. After the generation of the fuzzy association rules, the rules are extracted; Butterfly Optimization Algorithm based Artificial Bee Colony (BOAABC) algorithm is employed for the selection of optimal rules. At last, optimal rules employ different mechanisms, which rely on fuzzy support, confidence, and distributed training set coverage. The experimental results prove that the proposed system yields improved performance when matched with the available system with respect to metrics such as accuracy, precision, recall, f-measure and error rate.

REFERENCES

[1] W. Altaf, M. Shahbaz & A. Guergachi (2017), "Applications of Association Rule Mining in Health Informatics: A Survey", *Artificial Intelligence Review*, Vol. 47, No. 3, Pp. 313–340.

[2] M. Abdel-Basset, M. Mohamed, F. Smarandache & V. Chang (2018), "Neutrosophic Association Rule Mining Algorithm for Big Data Analysis", *Symmetry*, Vol. 10, No. 4.

[3] I. Qureshi, B. Mohammad & M.A. Habeeb (2019), "Optimizing Association Rule Mining using Walk Back Artificial Bee Colony (walkback abc) Algorithm," *Innovations in Computer Science and Engineering*, *Springer*, Singapore, Pp. 39–48.

[4] A. Segatori, A. Bechini, P. Ducange & F. Marcelloni (2017), "A Distributed Fuzzy Associative Classifier for Big Data", *IEEE Transactions on Cybernetics*, Vol. 48, No. 9, Pp. 2656–2669.

[5] J.C.W. Lin, T. Li, P. Fournier-Viger, T.P. Hong, J.M.T. Wu & J. Zhan (2017), "Efficient Mining of Multiple Fuzzy Frequent Itemsets", *International Journal of Fuzzy Systems*, Vol. 19, No. 4, Pp. 1032–1040.

[6] A.A. Zoraghchian, M.K. Sohrabi & F. Yaghmaee (2021), "Exploiting Parallel Graphics Processing Units to Improve Association Rule Mining in Transactional Databases using Butterfly Optimization Algorithm", *Cluster Computing*, Pp. 1–12.

[7] A. Telikani, A.H. Gandomi, A. Shahbahrami & M.N. Dehkordi (2020), "Privacy-preserving in Association Rule Mining using an Improved Discrete Binary Artificial Bee Colony", *Expert Systems with Applications*, Vol. 144, No. 11, Pp. 1–19.

[8] L.T. Nguyen, B. Vo, L.T. Nguyen, P. Fournier-Viger & A. Selamat (2018), "ETARM: An Efficient Top-k Association Rule Mining Algorithm", *Applied Intelligence*, Vol. 48, No. 5, Pp. 1148–1160.

[9] F. Wang, K. Li, N. Duić, Z. Mi, B.M. Hodge, M. Shafie-Khah & J.P. Catalão (2018), "Association Rule Mining based Quantitative Analysis Approach of Household Characteristics Impacts on Residential Electricity Consumption Patterns", *Energy Conversion and Management*, Vol. 171, No. 9, Pp. 839–854.

[10] H. Zheng, J. He, G.Huang, Y. Zhang & H. Wang (2019), "Dynamic Optimisation based Fuzzy Association Rule Mining Method", *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 8, Pp. 2187–2198.

[11] S. Neelima, N. Satyanarayana & P.K. Murthy (2017), "Optimization of Association Rule Mining using Hybridized Artificial Bee Colony (ABC) with BAT Algorithm", *IEEE 7th*

International Advance Computing Conference (IACC), Hyderabad, India, Pp. 831–834.

[12] G.K. Ameta (2017), "An Improved Association Rule Mining Approach to Reduce Iterations in Ant Colony Algorithm through Artificial Bee Colony Algorithm", *International Journal of Latest Transactions in Engineering and Science*, Vol. 1. No. 3, Pp. 1–7.

[13] H. Faris, I. Aljarah & S. Mirjalili (2018), "Improved Monarch Butterfly Optimization for Unconstrained Global Search and Neural Network Training", *Applied Intelligence*, Vol. 48, No. 2, Pp. 445–464.

[14] A.K. Dubey (2021), "Optimized Hybrid Learning for Multi Disease Prediction Enabled by Lion with Butterfly Optimization Algorithm", *Sādhanā*, Vol. 46, No. 2, Pp.1–27.

[15] J.R. Lambert & E. Perumal (2021), "Oppositional Firefly Optimization based Optimal Feature Selection in Chronic Kidney Disease Classification using Deep Neural Network", *Journal of Ambient Intelligence and Humanized Computing*, Pp. 1–12.