Computational Journal of Mathematical and Statistical Sciences
4(1), 17–40
DOI: 10.21608/cjmss.2024.298501.1059
https://cjmss.journals.ekb.eg/

Research article

Leveraging Ant Colony Optimization to Uncover Customer Characteristics for Churn Prediction

Ibrahim Al-Shourbaji[1*], Abdoh Jabbari[1], Shaik Rizwan[1], Mostafa Mehanawi[1], Phiros Mansur[1], and Mohammed Abdalraheem[2]

[1] Department of Electrical & Electronics Engineering, Jazan University, Jazan 45142, Saudi Arabia.
[2] Department of Computer Science, Jazan University, Jazan 45142, Saudi Arabia.

* **Correspondence:** alshourbajiibrahim@gmail.com

Abstract: Customer churn prediction is a critical task in the telecommunication (telecom) industry, where accurate identification of customers at risk of churning plays a vital role in reducing customer attrition. Feature selection (FS) is an integral part in Machine Learning (ML) models which aims to improve performance and reduce computational time (CT). This work optimizes Ant Colony Opti- mization (ACO) and its structure to empower its capability for customer churn prediction in the telecom industry. The effect of the ACO's hyper-parameters, like the pheromone value, heuristic information, pheromone decay factor, and the number of ants, in the optimization process are investigated. The optimization objective is measured by evaluating the prediction performance of selected features using the k-nearest neighbor classifier. Experiments are performed on three different open-source customer churn prediction datasets. The results are evaluated using several evaluation metrics and compared with three other optimization methods. The findings show that the optimized ACO performs is better than the other comparative methods. The Friedman and Holms test demonstrate that optimized ACO is stable and effective. This work suggests that selected optimal customer characteristics can be utilized to offer valuable insights and reduce churning rate.

Keywords: Ant colony optimization, churn prediction, feature selection, metaheuristic algorithms.
Mathematics Subject Classification: 46N10; 62F07; 60G25
Received: 22 June 2024; **Revised:** 1 August 2024; **Accepted:** 10 September 2024; **Online:** 12 October 2024.

1. Introduction

Over time, the telecom industry has experienced rapid growth that it has intensified rivalry among service providers. This has led to a significant loss of revenue due to churning. Churner consumers are those who switch from one service provider to another in the market [1]. To better understand customers' demands, telecom businesses use customer relationship management systems as an integrated approach in their strategic plan. These systems contain historical data about the company clients, which ML techniques can be then used to turn those data into useful knowledge [2]. In ML techniques, data preprocessing is essential, and FS is generally considered a foremost preprocessing step.

The method of eliminating redundant and irrelevant features to determine the Optimal Feature Subsets (OFS) is known as FS. The primary aim of FS is to choose the most important features without altering the original data representation and, thus, to select OFS that have the lowest redundancy and the highest relevance to the task. It is essential in a wide range of applications, including text mining, image recognition, spam detection, and finance [3, 4, 5, 6]. According to [7], the benefits of FS are as follows: (i) reduces the data required for learning process, (ii) improves prediction, (iii) learns knowledge that can easily be understood, and (iv) minimizes computation time (CT).

A large number of features in datasets demand investigation of Metaheuristic Algorithms (MAs), which can deal with FS effectively [8, 9, 10]. Grey wolf optimizer [11], bat algorithm [12], cuckoo search algorithm [13], Particle Swarm Optimization (PSO) [14], Firefly Algorithm (FFA) [15], dragonfly optimization algorithm [16], crow search algorithm [17], ACO [18], and Multi-Verse Optimizer (MVO) [19] are examples of well-known MAs. This class of algorithm has gained popularity in a wide range of applications [20]– [23].

Several researchers used ACO for classification in the application of customer churn prediction. In [24], ACO is applied to induce an accurate and comprehensible rule-based classification model. In another work [25], the author discussed the potential of utilizing ACO in the application of churn prediction and the results showed that the ACO attains a compelling performance. In [26], Mult objective cost sensitive ACO (MOCS-ACO) with genetic algorithm (GA) is combined to improve classification results. The GA is employed to select OFS while the MOCS-ACO is used as a classification model in [27], the authors employed ACO to identify OFS and the identified features are then fed to the gradient boosting tree (GBT) model. The results showed that the proposed ACO-GBT model produced good results in predicting customer churn. In [28], the author discussed the advantages of utilizing MA for churn prediction, specifically ACO for churn prediction. The findings confirmed that ACO is a powerful method for churn prediction. In [29], ACO is combined with RSA to boost its capacities as for churn prediction. Experimental results showed good performance of ACO-RSA. In [30], ACO-based feature selection is used to train gradient boosting tree for churn prediction,

Although the literature showed several works use ACO as a FS method in the application of customer churn prediction, a systematic investigation of the effect of hyper-parameter tunning on the ACO for this application is not available. The current work aims to fill this gap by improving customer churn prediction using ACO hyper-parameter tuning. The core contributions of this work can be summarized as follows:

- To investigate the effect of hyper-parameter tuning on the performance of ACO when it used as a FS method
- To assess the ACO improvement using three open-source customer churn prediction datasets
- To quantize improvements in the ACO's performance in terms of CT, fitness value, accuracy, and fraction of selected features
- To compare its efficiency and performance against other MAs

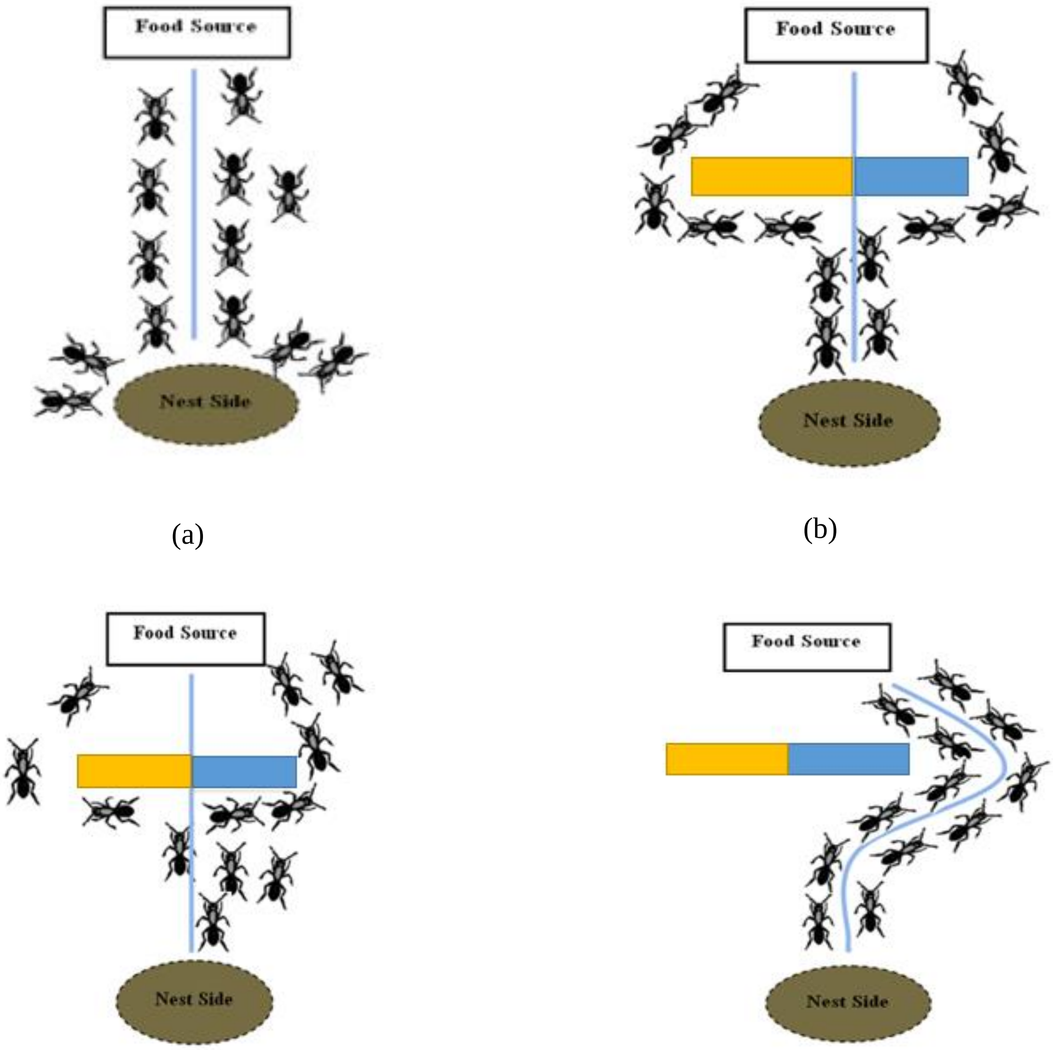
Section two provides a briefly describes process of ACO, followed by the description of three datasets for customer churn prediction in section three. The tuned ACO and its hyper-parameters are presented in section four. Experimental results and statistical comparison with other state-of-the-art FS methods are given in section five. Finally, section six concludes the work.

2. Ant Colony Optimization

Initially known as Ant System, ACO is a population-based MA [18]. The attributes of ACO enhance its efficacy compared to other MAs by facilitating parallel processing, sidestepping process dependency, and providing valuable feedback on the behaviours of ants during the search process [31]. It applied a minimal change to other combinative stochastic optimization, and it can also apply the same versions to the problem without extra modifications.

Ants try to find the shortest route between their nest and the food source. They leave a chemical material, known as pheromone, along their trail. The pheromone serves as a means of communication among the ants, indicating the shortest path to gather food. Initially, ants move randomly in search of food, smelling the chemical material deposited by ants with prior traversal along the path, thereby attracting other ants to follow. Therefore, the probability of that path will be increased since more ants traverse via that path [32].

Fig 1 explains the ants' behaviour while seeking food; the shortest path from the nest to the source is used for collecting food, as shown in Fig 1(a). The ants walk from the nest to the food source directly. Suddenly, an obstacle crosses the path with the yellow side longer than the blue side, as shown in Fig 1(b). The ants redirect the path to the food source. At first, ants walk in arbitrary directions and drop the pheromone on their way. The effort required to complete the path from the blue side is less than the yellow side; therefore, the pheromone quantity sensed by ants on the blue side is more than the yellow side. Thus, the number of ants walking through the blue side is more than the opposite side, which makes it more attractive for the next ants to follow, as shown in Fig 1(c). Finally, more ants travel the blue side, indicating the best-found solution and the preferable choice by the ants as the shortest path since it has a higher probability, as shown in Fig 1(d).



(a) (b) (c) (d)

Fig 1. Process of path finding for ants: (a) original path between the nest and the food source, (b) path after introducing an obstacle having a larger yellow side than blue, (c) when pheromone deposition on the blue side increases cumulatively, and (d) converged to the shortest path.

An ACO-based FS method typically involves a graph, as shown in Fig 2. A fully connected graph facilitates the movement of ants from any feature F_i to feature F_j with the relationship between the features indicated as w_{ij}. For making probabilistic decisions, ACO uses heuristic information (evaluated fitness) and pheromone trail (founded solution). While traversing a path, ants crossing a feature increase the pheromone level deposited at that feature. This boosting enhances the likelihood of the feature being in the shortest path. An example of ant traversing the graph is shown in Fig 2. An ant is initially positioned at feature F_1, it then chooses to move to F_2

and then F_4 based on the pheromone trail and heuristic information. Finally, upon arrival at F_4, the OFS: $\{F_1, F_2, F_4\}$ is generated.

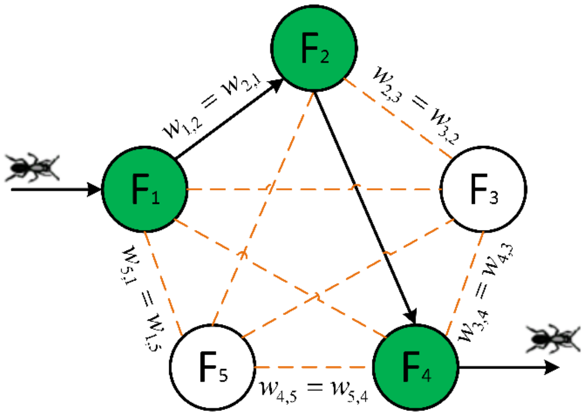


Fig 2. Fully connected graph representation of ACO for FS.

After randomly positioning ants on a set of features, the pheromone level is initialized $\tau_0 = 1$ with maximum number generations *G*. A combination of both pheromone level $\tau_i$and heuristic information η_i represented classifier accuracy of *i*th feature for a class label of *i*th feature forms the "transition matrix" to move to the next possible feature.

At every generation g, $TP_i^k(g)$ is the transition probability of *k*th ant at *i*th feature, and it is calculated as [33]:

$$TP_i^k(g) = \begin{cases} \dfrac{[\tau_i(g)]^\alpha [\eta_i]^\beta}{\sum_{j \in j_i^k} [\tau_i(g)]^\alpha [\eta_i]^\beta} & \text{if } j \in j_i^k \\ 0, & otherwise \end{cases} \qquad 1$$

where j_i^k is a neighbours of *i*th features that the *k*th ant does not visit. The non-negative hyper-parameters. β and α represent importance of heuristic information and pheromone level, respectively.

After selecting the next feature along the path, the updated set of selected features undergoes evaluation by a fitness evaluator. In this work, a k-nearest neighbor (KNN) with five neighbours is used to evaluate the quality of selected features due to their popularity [34] and the capability to measure distances between numerical or categorical attributes [35]. The fitness value of select features $S^k(g)$is calculated as:

$$\gamma(S^k(g)) = w(1 - accuracy) + (1 - w)\frac{|S^k(g)|}{N} \qquad 2$$

where *w* weights the classifier performance importance and the number of selected features $|S^k(g)|$.

In this work, we decided to give higher importance to classifier performance by setting w = 0.99. The movement of *k*th ants can be stopped when the minimum number of features is visited, or no the fitness value does not improve with a new feature [18]. In this work, a minimum number of selected features is set to half of the total number of features as stopping criteria to avoid retaining more than 50% of the original features in the dataset. At generation (g+1), pheromone level at ith feature is calculated as [36]:

$$\tau_i(g+1) = (1-p)\tau_i(g) + \sum_{k=1}^{K} \Delta\tau_i^k(g) \qquad 3$$

where,

$$\Delta\tau_i^k(g) = \begin{cases} \gamma\left(S^k(g)\right)/\left|S^k(g)\right|, & if\ i \in S^k(g) \\ 0, & otherwise \end{cases} \qquad 4$$

where the decay rate ($0 \leq p \leq 1$) controls pheromone evaporation, and $\gamma\left(S^k(g)\right)$ is the fitness value of the features subsets $S^k(g)$ found by kth ants at generation g. The kth ants' deposits pheromone $\Delta\tau_i^k$ at traverses' ith feature, else no pheromone is deposited.

The optimization stops when g reaches the maximum iterations G. Finally, when all the ants complete the path, the features are selected based on the smallest fitness value. The flow diagram process of the ACO as an FS method is shown in Fig 3.

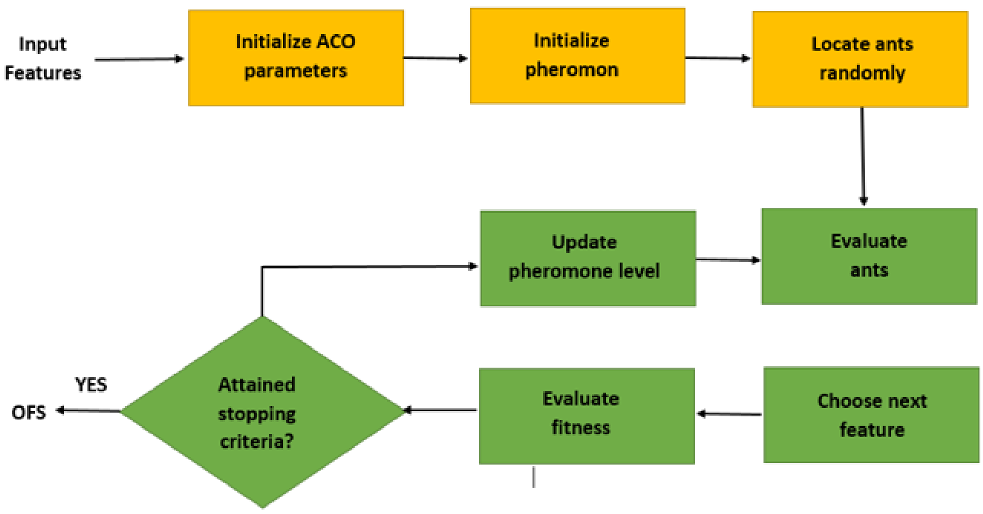


Fig 3. Flow diagram of ACO's general process

Algorithm 1. Pseudocode for ACO-based FS method

Begin
Initialize the parameters of the ACO: $\alpha, \beta, p, m, w, G$
Let $g = 1$
for each feature i **do**
    $\tau_i = \tau_0$
end for
Place m ants randomly on a set of features with an initial pheromone level
while g ≤ G **do**
    for each ant $k = 1, \ldots, m$ **do**

S^k= { }
while stopping criteria not reached **do**
from the current feature, select the next features using Eq. 1
add the selected feature to the subset S^k
end while
calculate fitness value using Eq. 2
end for
for each feature i **do**
Update pheromone level using Eq. 3 and 4
end for
g = g+1
end while
Return set of selected features S^k with highest $\gamma(S^k)$ as the OFS
End

3. Datasets

Three datasets from the application of customer churn prediction are used, and their characteristics are provided in Table 1.

Table 1. The characteristics of the datasets

Dataset	Number of instances	Number of features
[1] dataset-3	3333	21
[2] dataset 2	7043	21
[3] dataset-3	100,000	50

Datasets are reprocessed with the step-by-step procedure as follows:

I. The attributes that consist of unique values are ignored since they do not affect the model training process

II. Categorical values {yes and no}, {true and false} are into converted into binary.

III. Continuous values are normalization in the range 0–1 using a Linear transformation (L') as

[1] http://www.sgi.com/tech/mlc/db/

[2] https://www.ibm.com/analytics/us/en/

[3] https://www.kaggle.com/abhinav89/telecom-customer/data

$$L' = [(Lmin)/(max - min)] * (max' - min') + min' \quad (5)$$

where min is the old minimum value, min' is the new minimum, max is the old maximum, max' is new maximum.

4. ACO Hyper-Parameters Tuning

Tuning of hyper-parameters is a vital step to increase the performance of the ACO algorithm. ACO has several hyper-parameters, including the relative importance of the pheromone value (α), heuristic information (β), pheromone decay factor (p) and the number of used ants in the population. The tuned values of these hyper-parameters will be obtained by minimizing the number of Selected Features (SF) and CT using the datasets provided in Table 1.

4.1. dataset-3

4.1.1. Effect of α and β

The α and β together decide the transition probability for the ants' movements from one feature to another. In this work, we investigated a combination of $\alpha$and$\beta$for achieving the minimum number of SF and CT by varying each value over the range of 0–1.8, as shown in Fig 4. For lower values of α< 0.4, CT is very high. As the value of α > 1 increased, CT decreased with approximately a minimum of 6.03 seconds for α = 1.2 and β = 1.2, 1.3. The range of SF with varying α and β values is 2–12, as shown in Fig 4 (b). Hence, the optimum αvalues are within the range of 1.3-1.6. and β within the range of 0–1.5.

(a)

(b)

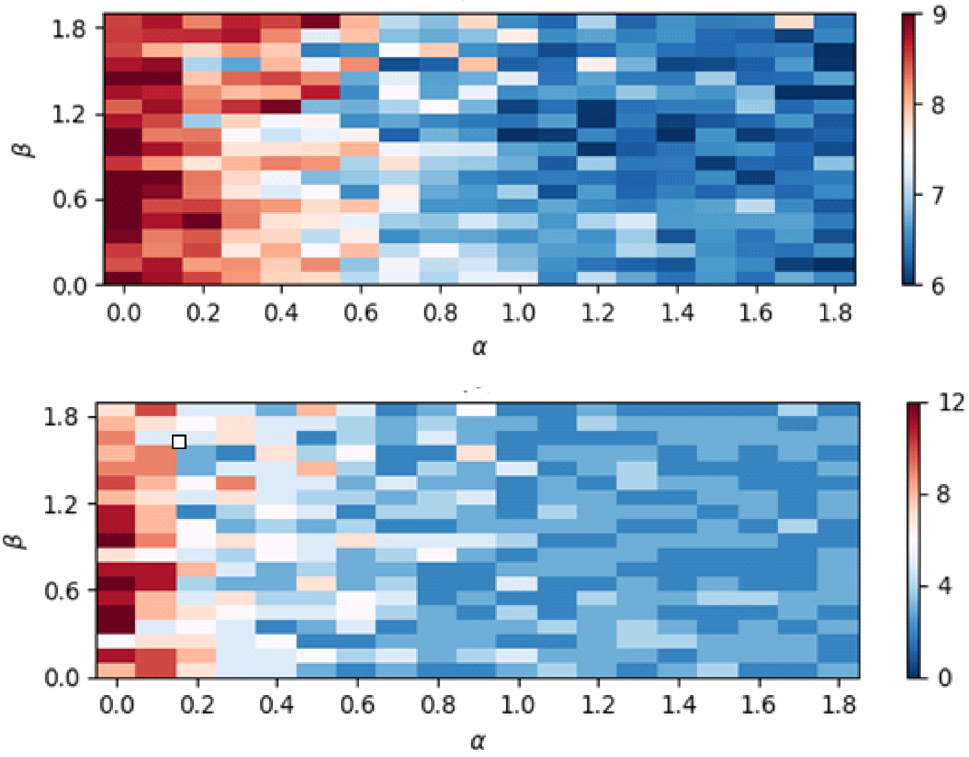


Fig 4. Effects of varying combinations of α and β values on (a) CT and (b) SF.

In order to choose optimal α and β values for the current task, we calculated the mean and standard deviation (std) of CT and SF using five generations for each value of α, as shown in Fig 5 [37]. It can be noted that α = 1.4 has the least mean and std. The optimum value of β = 1.2 is selected by searching for minimum CT and SF across the column α = 1.4. Hence, we choose the optimum values as α = 1.4 and β = 1.2.

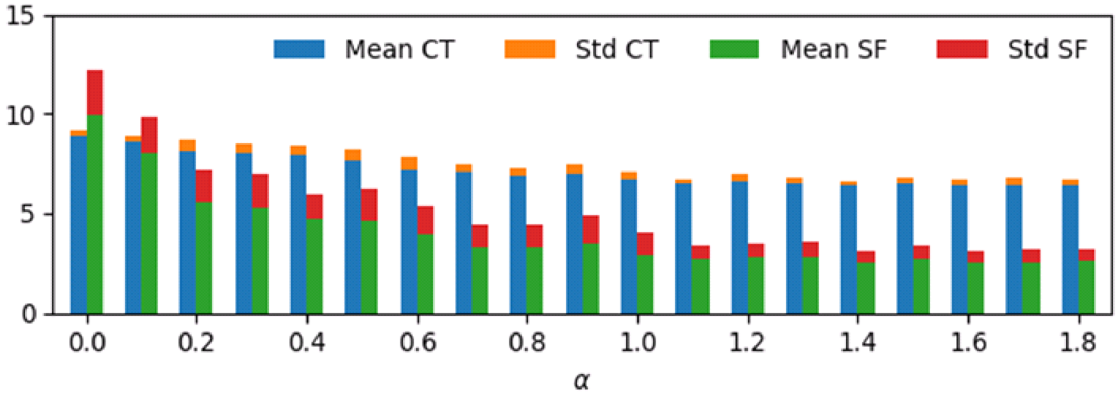
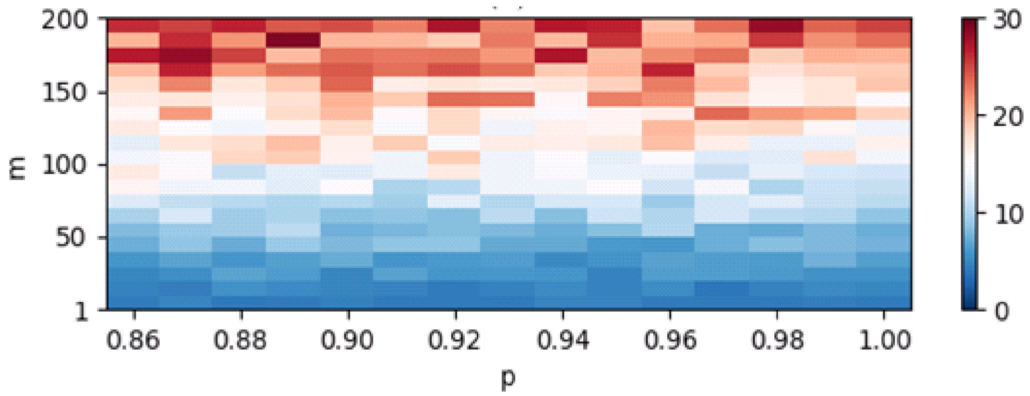


Fig 5. Mean and std of CT and SF for different α range.

4.1.2. Effect of number of used ants and pheromone decay factor

The minimum population *m* and *p* together decide the pheromone level on each path. The high number of m and low p will result in a high pheromone level on the optimum path but will increase CT. In this work, we investigated m and p combinations for the CT and the minimum SF, as shown in Fig 6. The CT for ACO with varying m and p over the range 1–200 and 0.86–1, respectively, are shown in Fig 6 (a). As estimated for the increasing number of *m*, the CT is also increasing significantly. Hence, the optimum solution must choose the minimum possible m. Fig 6 (b) shows SF with varying the number of ants and *p* in the range 4–16.

(a)



(b)

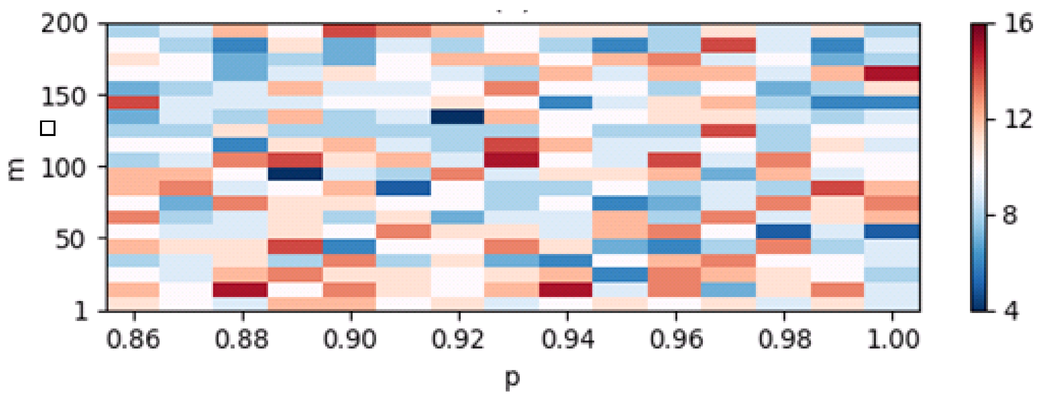


Fig 6. Effects of varying combinations of m and p on (a) CT and (b) SF.

To better decide optimum values for the m and p, we calculated the mean and std of SF for each value of p, as shown in Fig 7. It can be observed that p = 0.87, 0.95, and 0.98 have the smallest mean with std. However, p = 0.95 shows the least number of SF with the smallest possible number of m. Hence, we choose the optimum values m = 30 since accuracy does not increase by increasing m and p = 0.95 for dataset-3.

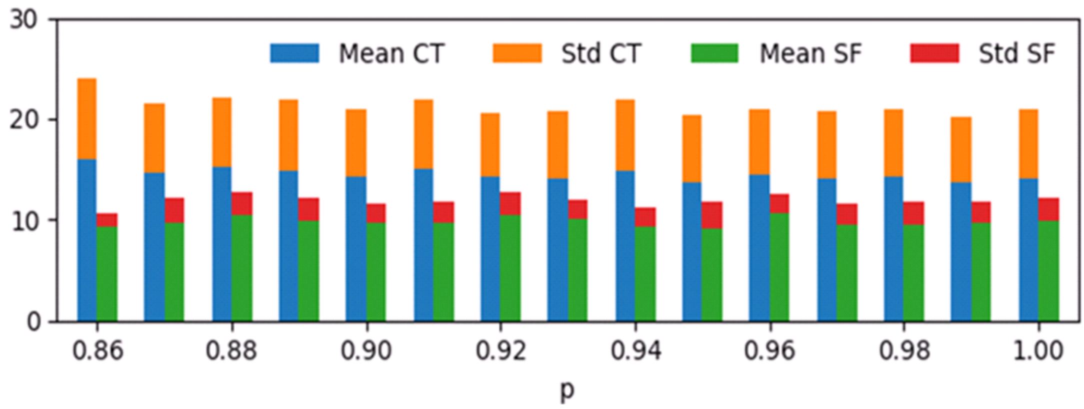


Fig 7. Mean and standard deviation of CT and SF for varying p.

The ACO optimum parameters for dataset-3 are provided in Table 2.

Table 2. ACO optimal parameters for dataset-3

Hyper-parameter	Value
α	1.4
β	1.2
p	0.95
m	30

4.2. dataset-3

4.2.1. Effect of α and β

Fig 8 shows the effects of α and β values on the CT and SF. It can be observed from Fig 8(a) that for lower values of $\alpha < 0.6$, CT is very high. As the value of $\alpha > 1$ increased, CT decreased with an approximate minimum of 5.03 seconds for $\alpha = 1.1$ and $\beta = 1.7$. The range of SF with varying α and β values is 4–16, as shown in Fig 8 (b). Hence, the optimum αvalues are within the range of 1.2–1.6. and β within the range of 0–1.5.

(a)

(b)

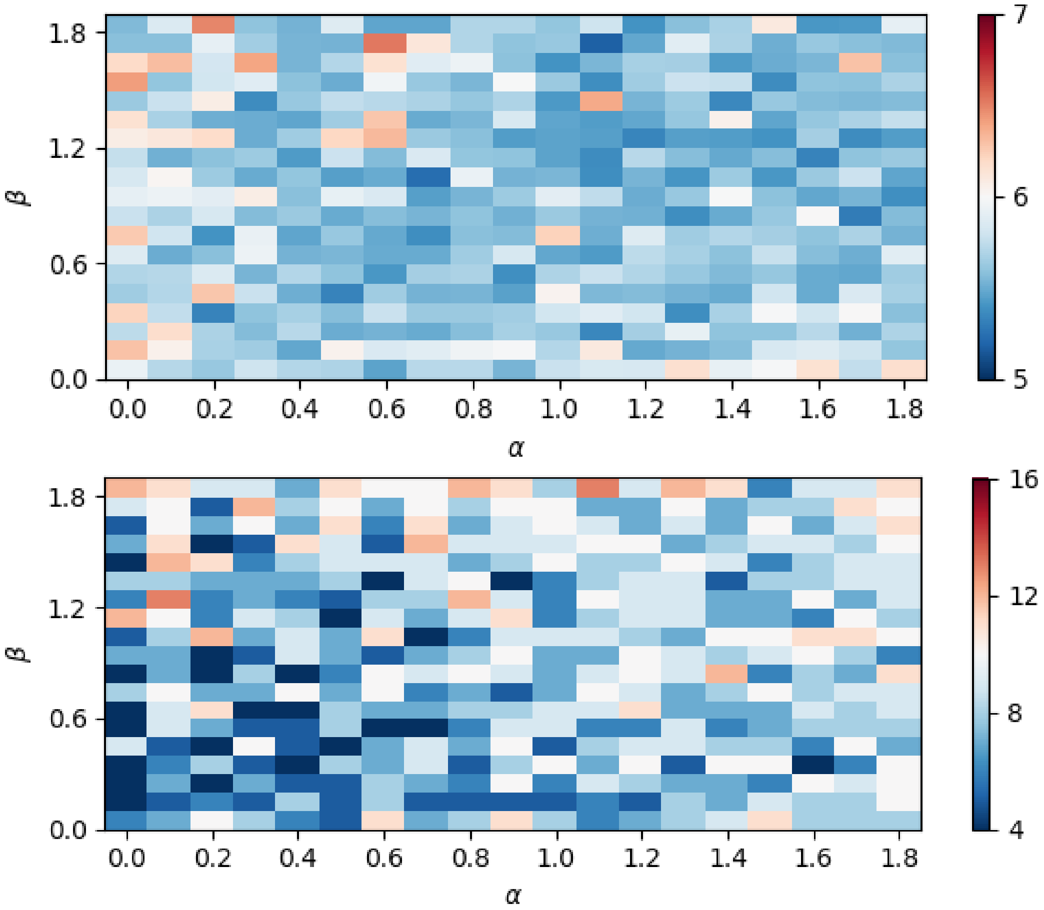


Fig 8. Effects of varying combinations of α and β values on (a) CT and (b) SF.

The mean and std of the CT and SF using five generations were calculated for each value of α as shown in Fig 9. It can be noted that $\alpha = 1.2$ has the least mean and std. The optimum values of α andβ are 1.2 and 1.2, respectively, for dataset-3.

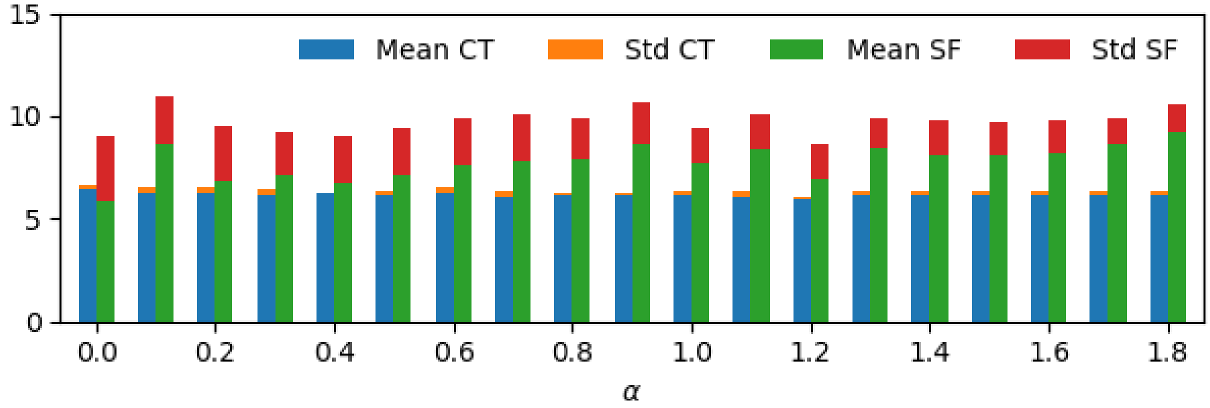
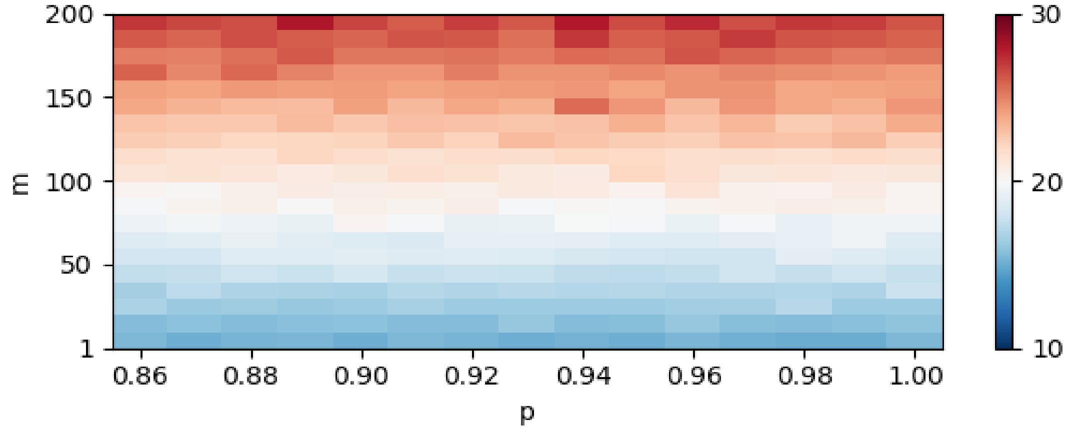


Fig 9. Mean and std of CT and SF for different αrange.

4.2.2. Effect of number of used ants and pheromone decay factor

The CT for ACO with varying m and $p$over the range 1–200 and 0.86–1, respectively, as shown in Fig 10 (a). As estimated for the increasing number of m, the CT also increased. Fig 10 (b) shows SF with varying the number of ants and p in the range 4–16.

(a)



(b)

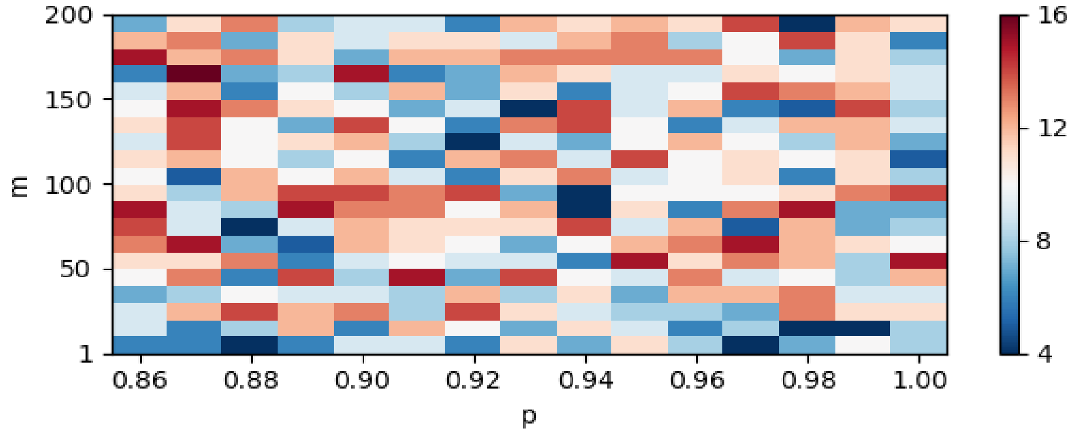


Fig 10. Effects of varying combinations of m and p (a) CT and (b) SF.

It can be observed from Fig 11 that p = 0.94 has the smallest mean and std. and the optimum value of m = 60 is selected indataset-2. The tuned ACO hyper-parameters for dataset-3 are provided in Table 3.

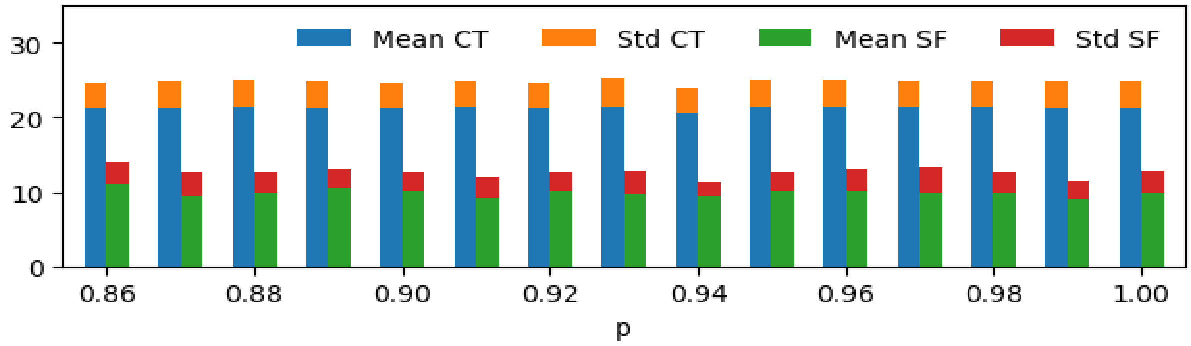


Fig 11. Mean and standard deviation of CT and SF for varying p.

Table 3. ACO optimal parameters for dataset-3

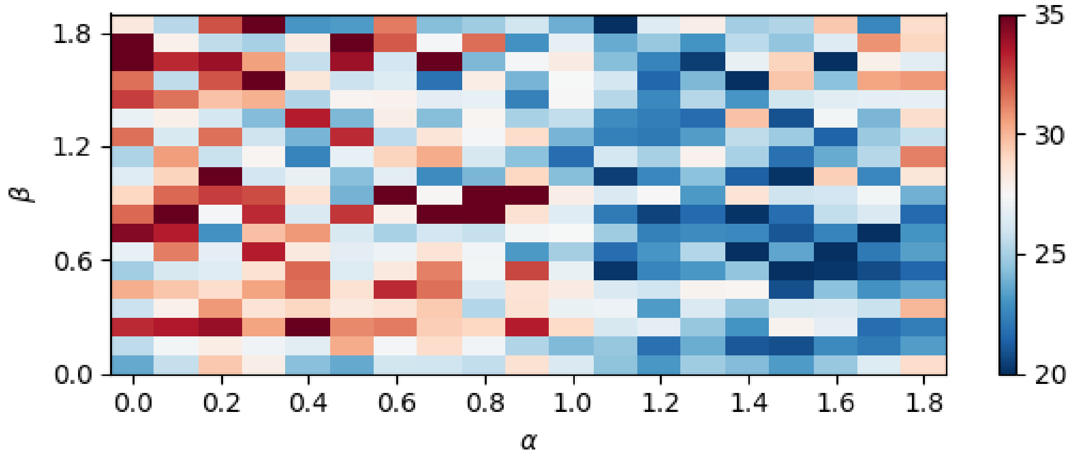
Hyper- parameter	**Value**
α	1.2
β	1.2
p	0.94
m	60

4.3. dataset-3

4.3.1. Effect of α and β

Fig 12 shows the effects of α and β values on the CT and SF. From both Fig 12 (a) and (b), the optimum αvalues are within the range of 1.1–1.3. and β within the range of 0–1.5, respectively.

(a)



(b)

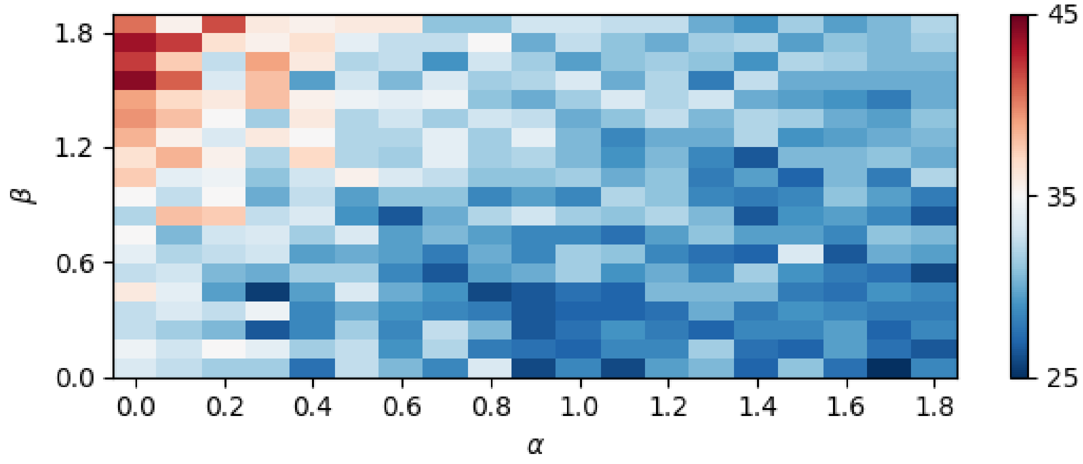


Fig 12. Effects of varying combinations of α and β values on (a) CT and (b) SF.

It can be observed from Fig 13 that $\alpha = 1.2$ had the least mean and std. The optimum value of $\beta = 0.8$ is selected by searching for minimum CT and SF across the column $\alpha = 1.2$. Hence, the optimum values as$\alpha = 1.2$ and$\beta = 0.8$ are selected.

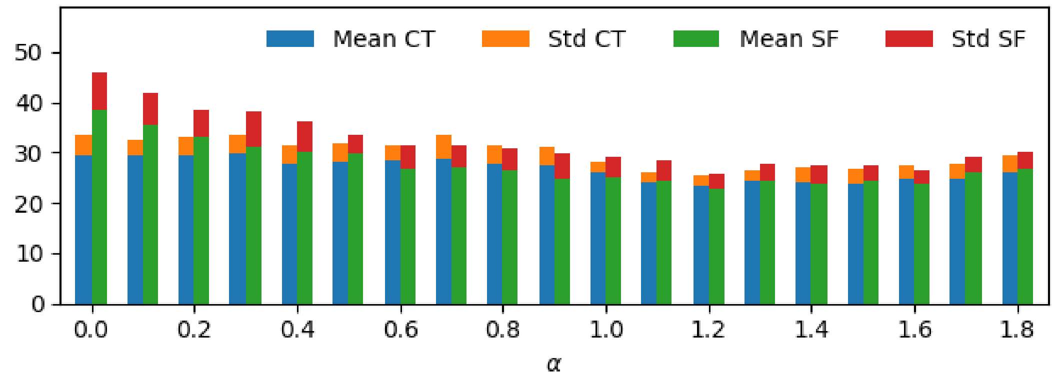
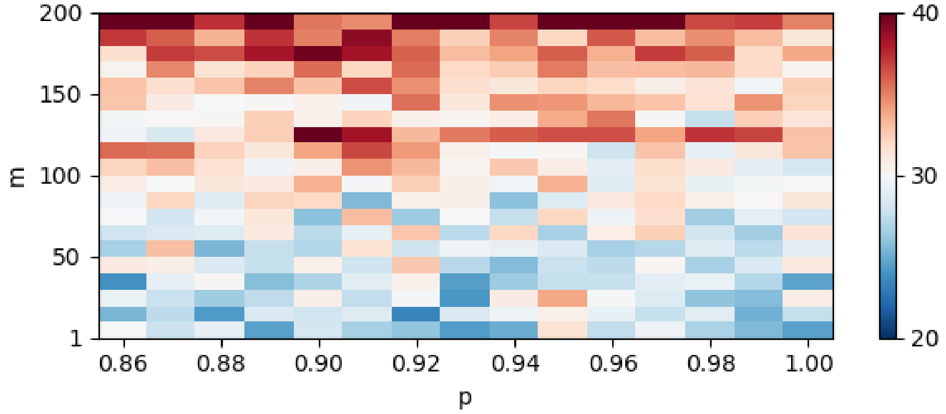


Fig 13. Mean and std of CT and SF for different α values.

4.3.2. Effect of number of used ants and pheromone decay factor

The CT for ACO with varying m and p over the range 1–200 and 0.86–1, respectively, are shown in Fig 10 (a). As estimated for the increasing number of m, the CT is also increasing. Fig 14(b) shows SF with varying the number of ants and p in the range of 25–45.

(a)



(b)

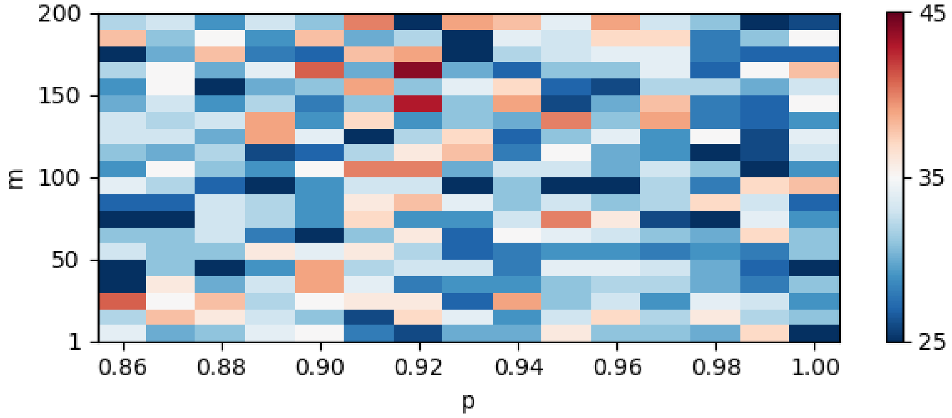


Fig 14. Effects of varying combinations of m and p on (a) CT and (b) SF.

In Fig 15, p = 0.93 has the smallest mean and std as shown. However, p = 0.93 shows the least number of SF with the smallest possible number of m. Hence, we choose the optimum values m = 90 since accuracy does not increase by increasing m and p = 0.93 for dataset-3. The ACO optimum parameters are provided in Table 4for dataset-3.

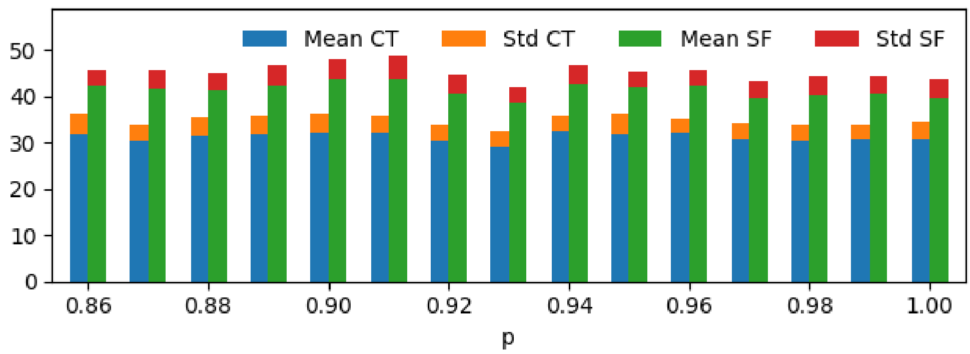


Fig 15. Mean and standard deviation of CT and SF for varying p.

Table 4. ACO optimal parameters for dataset-3

Hyper-parameters	Value
α	1.2
B	0.8
p	0.93
m	90

5. Results and discussion

In order to examine the effectiveness of the ACO as a FS method, the datasets provided in Table 1 are used and compared against other MAs. These comparative algorithms include PSO, MVO and FFA. Each dataset is divided randomly, where 50 % of the samples are used as a training dataset and the rest as a testing dataset. The ACO settings parameters provided in Tables 2, 3 and 4 are used for each dataset, and the maximum number of generations is set to G=100 for both the ACO and the other comparative methods. Then, the OFS obtained by the ACO and the other comparative methods from the training dataset are used as inputs to train and test SVM with radial basis function kernel (SVM_{rbf}) classifier. The models are executed using Python on Windows 10 machine with 32 GB RAM and 3.13 GHz processor.

ACO and the other comparative methods are run 20 independent times as recommended by [30], and the average results are calculated. The results of the ACO and the other comparative methods in terms of accuracy are given in Table 4. The results of the number of features along with the OFS are provided in Table 5, while the results in terms of fitness values and CT are shown in Table 6. Accuracy, a fraction correctly classified test cases by a classifier, can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad 6$$

where True Positive (TP) and True Negative (TN) denote correctly identified churners and non-churners. False Positive (FP) is incorrectly classified churner and False Negative (FN) is incorrectly classified non-churner.

Table 5 Average performance results in terms of accuracy

Dataset	PSO	MVO	FFA	ACO
taset-1	0.7251	0.7523	0.7723	0.8200
taset-2	0.6675	0.7566	0.7538	0.7720
taset-3	0.5322	0.5681	0.5838	0.6030

Table 6 The number of features and OFS for the comparative methods

Dataset	PSO	MVO	FFA	ACO
taset-1	12 (OFS:1,3,5,8,10,11,13,14,15,16,17,18)	5 (OFS:4,5,10,13,15)	9 (OFS: 2,3,4,7,10,14,15,17,18)	9 (OFS: 2,3,4,8,11,13,14,16,17)
taset-2	8 FS:2,8,11,13,16,17,18, 19)	6 OFS:2,5,7,11,12, 13)	6 FS:8,11,13,15, 18,19)	6 OFS:2,9,10,15,16,19)
taset-3	5 (OFS:1,4,6,7,8,10, 11,14,16,17,18,19, 20,21,23,25,27,28, 29,31,32,33,35,36, 37,38,40,41,42,43, 44, 46,47,48,49)	2 (OFS:1,4,6,7,8,10, 11,13,14,15,16,17, 18,19,21,22,24,28, 29,30,31,32,35,38, 40,41,43,44,45,46, 47,49)	30 (OFS:1,4,5,6,7,810,11,16,17,18, 19, 20, 21,22,24,29,31,32, 33,35,37,40,41,42, 44,45,46,47,49)	25 (OFS:1,3,4,7,10,11,12,13,14,19,20,24, 27,31,32,35,36,39, 40,42,43,44,45,46, 47)

Table 7. Average performance results in terms of fitness for the comparative methods

Dataset	PSO	MVO	FFA	ACO
dataset-1	0.2819	0.2526	0.2343	0.1861
dataset-2	0.3397	0.2487	0.2515	0.2332
dataset-3	0.4795	0.4427	0.4264	0.4060

Table 8. Average performance results in terms of CT (in seconds) for the comparative methods

Dataset	PSO	MVO	FFA	ACO
dataset-1	14.9085	14.3480	15.2210	11.2210
dataset-2	28.5087	28.5443	29.2910	23.2210
dataset-3	311.999	296.788	282.170	233.349

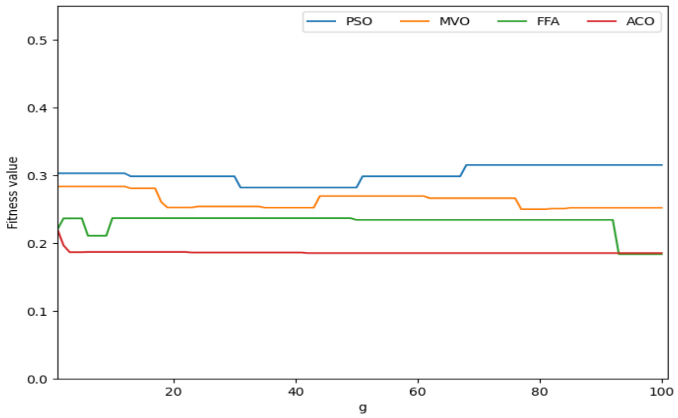
In Table 5, the ACO gained the best average accuracy results compared to the other comparative FS methods in all the used datasets. Also, the features selected using different FS algorithms are shown in Table 6. The number of features selected by ACO is smaller than other comparative methods. Although the number of selected features is almost the same, the actual selected features are different because of the difference in FS methods.

The ACO outperformed the other methods in terms of fitness values with 0.1861, 0.2332, and 0.4060 for datasets 1–3, respectively, as shown in Table 7. This is mainly because the ACO is capable of balancing between exploration (i.e., pheromone level) and exploitation (i.e., heuristic information), which can be attained through the proper selection of α and β parameters.

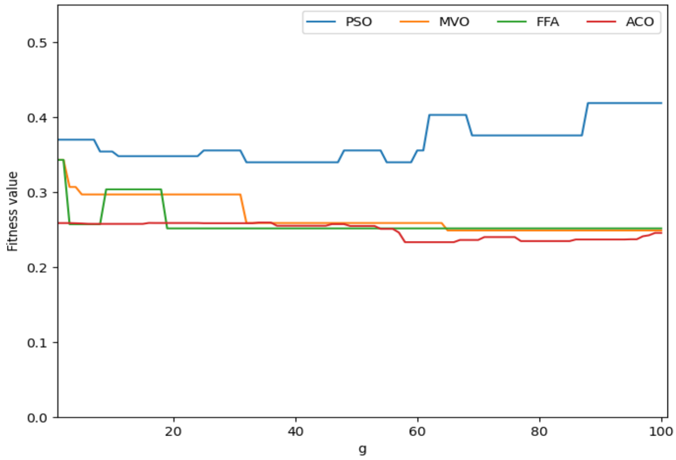
The ACO gets the lowest average CT among the other methods, which can be observed in Table 8. The CT is directly proportional to the number of examples and features. For example, MAs take higher CT due to more features and examples in dataset-3. However, the ACO takes lower CT compared to PSO, MVO and FFA methods because each ant in the ACO that passes optimal features is tagged so that other ants are attracted to follow the same path and therefore, the required CT to search all the feature space is minimized. As a result, the ACO reduces CT, and it can outperform the other methods in all the datasets used.

Convergence behaviour is used to better understand the behaviour of the FS algorithms. Fig 16 shows the convergence behaviour of all the methods, variation of fitness values with the number of generations, after 20 independent runs of MAs for each dataset. It can be observed in Fig 16 that the ACO has fast convergence speed and good local extremum avoidance proved by convergence analysis than the other methods. Also, ACO is constant after a few generations, indicating the robustness in selecting a maximum number of generations and avoiding local minima problem. For a small number of generations, PSO and FFA obtained good fitness values, while MVO get stuck in local minima problems.

(a)



(b)



(c)

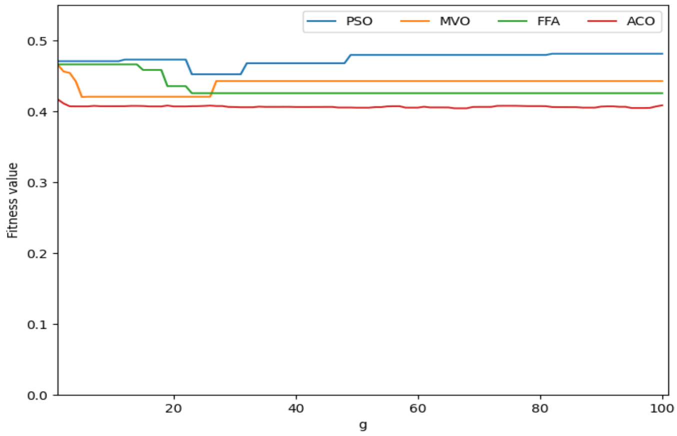


Fig 16. The convergence behavior plots: (a) dataset-1, (b) dataset-2, and (c) dataset-3

The results in terms of the average accuracy and the FS methods using the datasets are shown in Fig 17. It can be observed from the box plots in Fig 17 that the generated dispersion degree (i.e., the spacing between the best, median, and worst) by the ACO method is lower than the other methods in all datasets, which indicates its stability compared to the other methods. This is because the parameters optimization for each dataset allows ACO to have a better balance between exploration and exploitation. This confirms the superiority of ACO over the other MAs.

(a)

(b)

(c)

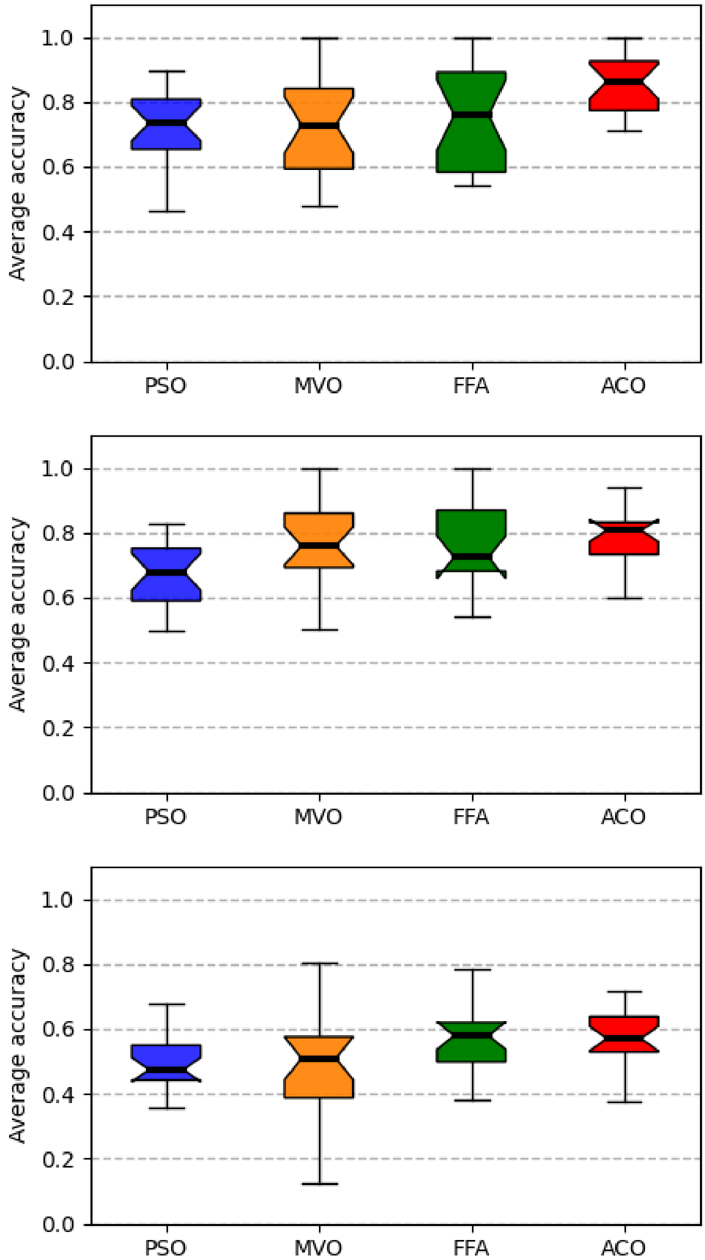


Fig 17. Box-plot for accuracy of the methods for (a) dataset-1, (b) dataset-2 and (c) dataset-3

5.1. Statistical significance

To further compare performance of the ACO and other comparative methods, Friedman's statistical test is employed to compute average ranking. The null hypothesis asserts that the comparative methods exhibit equal behaviour, while the alternative hypothesis suggests otherwise. Fig 18 illustrates the average ranking for all the methods using the accuracy results provided in Table 4, and the ACO is selected as the control method.

Friedman's test calculates the p-value as 7.9405E-7, indicating a substantial difference between the other methods. The results in Fig 18 validate that the ACO significantly performs better than PSO, MVO and FFA since it has the lowest rank.

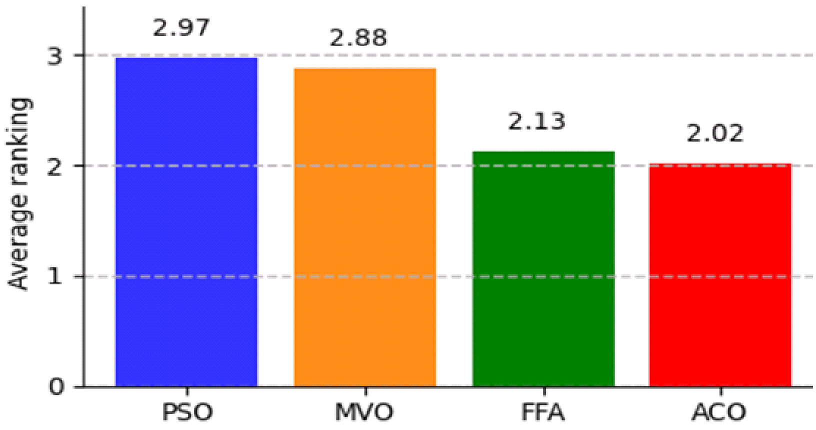


Fig 18. Average rankings of FS methods using the Friedman's test

Here, the rankings computed by the Friedman test are used and the adjusted p-value between the ACO and other comparative methods is calculated by using Holm's method, as given in Table 9. From this table, it can be seen that there is a significant difference in the behaviour between the ACO, PSO, and MVO, while there is no significant difference in behaviour between the ACO and FFA. This confirms the capability of the tuned ACO as a FS method for the application of customer churn prediction.

Table 9. Holm's method results among the ACO and other FS methods

nk	**;orithm**	**'alue**	**justed *p*-value**	**pothesis**
	Ͻ	000	000	ected
	/O	008	041	ected
	A	532	595	t-rejected

6. Conclusion and future work

FS is a typical problem in ML, and it is concerned with determining discriminating salient and redundant features from the whole set of features in a given dataset. This paper presents an improved FS method based on ACO. The effect of the ACO's hyper-parameters: Pheromone value, heuristic information, pheromone decay factor, and the number of ants, on the optimization process is tested. The optimization objective is measured by evaluating the prediction performance of selected features using the KNN. The efficiency of the improved method is evaluated using three open-source datasets for customer churn prediction. The performance of the improved method is also compared with three other FS methods, namely, PSO, MVO and FFA. Results showcased the superior capabilities of the tuned ACO in terms of fitness values, CT, accuracy, and selected features, providing robust and statistically significant results than the other FS methods. In future work, we will attempt to use tuned ACO and test its effectiveness as a FS method in other different applications such as renewal energy, signal processing and big data.

Funding: Not Applicable

Conflict of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Data availability statement: Data is available from the authors upon reasonable request.

References

[1] Sikri, A., Jameel, R., Idrees, S. M., & Kaur, H. (2024). Enhancing customer retention in telecom industry with machine learning driven churn prediction. Scientific Reports, 14(1), 13097.

[2] Wagh, S. K., Andhale, A. A., Wagh, K. S., Pansare, J. R., Ambadekar, S. P., & Gawande, S. H. (2024). Customer churn prediction in telecom sector using machine learning techniques. Results in Control and Optimization, 14, 100342.

[3] Manochandar, S., &Punniyamoorthy, M. (2018). Scaling feature selection method for enhancing the classification performance of Support Vector Machines in text mining. Computers & Industrial Engineering, 124, 139-156.

[4] AlShourbaji, I., Helian, N., Sun, Y., & Alhameed, M. (2021). Customer churn prediction in telecom sector: A survey and way a head. International Journal of Scientific & Technology Research (IJSTR).

[5] Hassani, Z., Hajihashemi, V., Borna, K., &SahraeiDehmajnoonie, I. (2020). A Classification Method for E-mail Spam Using a Hybrid Approach for Feature Selection Optimisation. Journal of Sciences, Islamic Republic of Iran, 31(2), 165-173.

[6] Liang, D., Tsai, C. F., & Wu, H. T. (2015). The effect of feature selection on financial distress prediction. Knowledge-Based Systems, 73, 289-297.

[7] Hall, M. A. (1999). Correlation-based feature selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, Hamilton

[8] Boussaïd, I., Lepagnot, J., &Siarry, P. (2013). A survey on optimization metaheuristics. Information sciences, 237, 82-117.

[9] Zhang, H., & Sun, G. (2002). Feature selection using tabu search method. Pattern recognition, 35(3), 701-711.

[10] Bermejo, P., Gámez, J. A., &Puerta, J. M. (2011). A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recognition Letters, 32(5), 701-711.

[11] Al-Tashi, Q., Rais, H. M., Abdulkadir, S. J., Mirjalili, S., &Alhussian, H. (2020). A review of grey wolf optimizer-based feature selection methods for classification. Evolutionary Machine Learning Techniques, 273-286.

[12] Rani, A. S. S., & Rajalaxmi, R. R. (2015, February). Unsupervised feature selection using binary bat algorithm. In 2015 2nd International conference on electronics and communication systems (ICECS) (pp. 451-456). IEEE.

[13] Salesi, S., &Cosma, G. (2017, October). A novel extended binary cuckoo search algorithm for feature selection. In 2017 2nd international conference on knowledge engineering and applications (ICKEA) (pp. 6-12). IEEE.

[14] Zhang, Y., Gong, D., Hu, Y., & Zhang, W. (2015). Feature selection algorithm based on bare bones particle swarm optimization. Neurocomputing, 148, 150-157.

[15] Zhang, L., Mistry, K., Lim, C. P., &Neoh, S. C. (2018). Feature selection using firefly optimization for classification and regression models. Decision Support Systems, 106, 64-85.

[16] Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., &Mirjalili, S. (2017, October). Binary dragonfly algorithm for feature selection. In 2017 international conference on new trends in computing sciences (ICTCS) (pp. 12-17). IEEE.

[17] Ouadfel, S., & AbdElaziz, M. (2020). Enhanced crow search algorithm for feature selection. Expert Systems with Applications, 159, 113572.

[18] Aghdam, M. H., & Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. IJ Network Security, 18(3), 420-432.

[19] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27(2), 495-513.

[20] Rajput, U., &Kumari, M. (2017). Mobile robot path planning with modified ant colony optimization. International Journal of Bio-Inspired Computation, 9(2), 106-113.

[21] Manjhi, Y., & Dhar, J. (2016). Forecasting energy consumption using particle swarm optimization and gravitational search algorithm. In 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), (pp. 417-420. IEEE.

[22] Chrouta, J., Chakchouk, W., Zaafouri, A., &Jemli, M. (2018). Modeling and control of an irrigation station process using heterogeneous cuckoo search algorithm and fuzzy logic controller. IEEE Transactions on Industry Applications, 55(1), 976-990.

[23] Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). Metaheuristic research: a comprehensive survey. Artificial Intelligence Review, 52(4), 2191-2233.

[24] Verbeke, W., Martens, D., Mues, C., &Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Systems with Applications, 38(3), 2354-2364

[25] Özmen, M., Aydoğan, E. K., Delice, Y., &Toksarı, M. D. (2020). Churn prediction in Turkey's telecommunications sector: A proposed multiobjective–cost-sensitive ant colony optimization. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(1), e1338.

[26] Özmen, M., Aydoğan, E. K., Delice, Y., & Toksarı, M. D. (2022). Churn prediction in Turkey's telecommunications sector: A proposed multiobjective–cost-sensitive ant colony optimization. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(1), e1338.

[27] Venkatesh, S., & Jeyakarthic, M. (2020). Metaheuristic based optimal feature subset selection with gradient boosting tree model for IoT assisted customer churn prediction. Journal of Seybold Report ISSN NO, 1533, 9211.

[28] Sivasankar, K. (2016). Effective Customer Churn Prediction on Large Scale Data using Metaheuristic Approach. Indian J. Sci. Technol, 9, 33.

[29] Al-Shourbaji, I., Helian, N., Sun, Y., Alshathri, S., & Abd Elaziz, M. (2022). Boosting ant colony optimization with reptile search algorithm for churn prediction. Mathematics, 10(7), 1031.

[30] AlShourbaji, I., Helian, N., Sun, Y., Hussien, A. G., Abualigah, L., & Elnaim, B. (2023). An efficient churn prediction model using gradient boosting machine and metaheuristic optimization. Scientific Reports, 13(1), 14441.

[31] Dorigo, M., Maniezzo, V., &Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.

[32] Wu, Y., Gong, M., Ma, W., & Wang, S. (2019). High-order graph matching based on ant colony optimization. Neurocomputing, 328, 97-104.

[33] Dorigo, M., Birattari, M., &Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[34] Dorigo, M., & Stützle, T. (2019). Ant colony optimization: overview and recent advances, pp. 311-351. Springer International Publishing.

[35] Guo, G., Neagu, D., & Cronin, M. T. (2005, August). Using kNN model for automatic feature selection. In International Conference on Pattern Recognition and Image Analysis , pp. 410-419. Springer, Berlin, Heidelberg.

[36] Wang, A., An, N., Chen, G., Li, L., &Alterovitz, G. (2015). Accelerating wrapper-based feature selection with K-nearest-neighbor. Knowledge-Based Systems, 83, 81-91.

[37] Rashidy, H. K., Faez, K., & Taheri, S. M. (2007). Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System. In Advances in Data Mining. Theoretical Aspects and Applications: 7th Industrial Conference, ICDM (pp. 14-18).

SASAR
Open Access Journal