The Scientific Temper (2024) Vol. 15 (spl): 379-388
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.44
https://scientifictemper.com/

RESEARCH ARTICLE

A metaheuristic optimisation algorithm-based optimal feature subset strategy that enhances the machine learning algorithm's classifier performance

V. Seethala Devi*, N. Vanjulavalli, K. Sujith and R. Surendiran

Abstract

In machine learning feature selection is a powerful stage of choosing a subset of features that are useful to increase performance while decreasing dimensionality. The rule of thumb in selecting feature subsets in classifiers is proposed in this paper using a new metaheuristic optimization algorithm, which intends to enhance classifier performance. The proposed method takes advantage of metaheuristic algorithms to better search and select the most important features that contribute to increasing classification performance, decreasing overfitting and increasing of speed of computation. We coordinate the optimization process with the diverse machine learning classifiers such as SVM, Random Forests, and Neural Networks to compare the performance of the chosen feature subsets. The current gist of the paper shows that benchmark results on suitable datasets show the outperformance of the proposed strategy over regular feature selection procedures, hence leading to enhanced classifier performance. Therefore, this research forms part of the existing knowledge in feature selection for improving classification performances in various machine learning algorithms by offering a reliable approach for determining and applying the best relevant features.

Keywords: Metaheuristic Optimization, Feature Selection, Machine Learning, Classifier Performance, Dimensionality Reduction, Support Vector Machines, Random Forests, Neural Networks.

Introduction

In today's big data environment, machine learning concerns numerous approaches to accurately identify relations and forecasts for different areas of application: medicine, banking, and construction, etc. With the increase in the size of datasets as well as their dimensions, there exists high possibility that there may be many features of the dataset that are either irrelevant or even noisy. Some may even have a negative impact on the model. This is a well-known problem that is often colloquially referred to as the curse of dimensionality, which, among other things leads to increased computational complexity and, higher level of overfitting and reduced classification performance. Feature selection, the process of choosing the relevant features required for the construction of the model, has, therefore, become critical in enhancing the model's efficiency and accuracy, Kanagarajan, S., & Ramakrishnan, S. (2018).

Feature selection is a process that tries to select relevant features, thus decreasing the dimensionality of data to make its learning easier for classification algorithms and improve their accuracy of classification. Traditional feature selection methods are commonly categorized into three types: filter, wrapper and embedded approaches. Filter methods quantitatively rank the features and pick the features using statistical possibilities regardless of the learning algorithm utilized. In turn, wrapper methods utilize the performance of a definite model for the assessment of subsets of characteristics and appear to be more accurate; however, they are considerably more computationally intense. Embedded methods adapt the procedure of selecting features with training models but may not

Research Department of Computer Science, Annai College of Arts & Science (Affiliated to Bharathidasan University, Tiruchirappalli), Kovilacheri, Kumbakonam, Tamil Nadu, India.

***Corresponding Author:** V. Seethala Devi, Research Department of Computer Science, Annai College of Arts & Science (Affiliated to Bharathidasan University, Tiruchirappalli), Kovilacheri, Kumbakonam, Tamil Nadu, India, E-Mail: seethaladevi1995@gmail.com

How to cite this article: Devi, V.S., Vanjulavalli, N., Sujith, K., Surendiran, R. (2024). A metaheuristic optimisation algorithm-based optimal feature subset strategy that enhances the machine learning algorithm's classifier performance. The Scientific Temper, **15**(spl):379-388.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.44

Source of support: Nil

Conflict of interest: None.

Received: 20/09/2024 **Accepted:** 15/10/2024 **Published:** 22/10/2024

effectively model interactions between features. While each method has its implications, these approaches often fail to address big-sized large datasets or control the exploration of computational efficiency trade-offs, Kanagarajan, S., & Ramakrishnan, S. (2015, December), Kanagarajan, S., & Ramakrishnan, S. (2016).

Metaheuristic optimization algorithms are optimization techniques based on the simulation of the biological world, which have proved useful in feature selection because of their flexibility and ability to operate in large search spaces. Unsupervised search methods, including genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO) all provide reliable platforms for getting close to the best features without cart-searching the universe. These algorithms tend to be very good at avoiding the sub-peak trappings and obtaining global solutions, making them ideal, especially for high-dimensionality features. This reveals that with the application of metaheuristic optimization into feature selection, machine learning models not only gain high performance but they also perform optimized computations, Kanagarajan, S., & Nandhini. (2020).

Literature Review

Feature selection plays a crucial role in most machine learning processes due to the fact that feature choice has a straight bearing on model performance, time complexity, and model interpretability where necessary. Significant work has been done in proposing methods of feature selection; these may be generally classified as filter, wrapper and embedded methods. More recently, metaheuristic optimization algorithms have been advocated and utilized to perform feature selection since they possess considerable abilities to search through large feature spaces to identify compact sets of features C. Arulananthan., & Kanagarajan, S. (2023).

Filter Methods

They construct separate measures for choosing features with reference to the particular learning algorithm as they do not have links with any learning algorithm but are mutually exclusive filter techniques. Such methods as Information Gain, Chi-Squared tests and Mutual Information are employed to rank and select the features. These methods are efficient from the computational viewpoint and can handle a large number of dimensions, which is always desirable for a method used in high-dimensional cases. Nevertheless, because they assess each feature independently, filter methods can ignore the interactions and relations among features that are crucial in classification processes. In any case, filter methods tend to provide sub-optimal subsets where features are related, C. Arulananthan et al. (2023).

Wrapper Methods

Wrapper methods work by assessing the feature subsets using the performance of a particular learning algorithm and, therefore are considerably better for the selection of features than filter methods. SFS, SBE and RFE are examples of methods under this category Squared-penalized likelihood coefficient, Robust Tuckerized rule, least-squares, coefficient of determination and Kulback-Leibler information are all examples of methods under this category. As a positive effect, wrapper methods allow using a priori knowledge of feature dependencies and interactions affecting model performance, but at the same time, these approaches are highly computationally expensive, particularly in the case of many features or their dimensions. Moreover, wrapper methods may suffer from overfitting problems since they depend on a certain model for feature selection, Vanjulavalli, N., Saravanan, M., & Geetha, A. (2016).

Embedded Methods

Integrated techniques include a feature selection step that is used during the model construction and is situated between filter and wrapper approaches. LASSO and Decision Trees remove features from the dataset naturally because features that are least important do not significantly affect the model. These methods present enhanced computational usefulness by making feature selection an element of the learning process and are generally much less sensitive to overfitting as when applying the wrapper methods. Nonetheless, embedded methods face certain limitations, where more specific methods of feature selection based on the model may have a limited use when projected for different classifiers, Vanjulavalli, D. N., Arumugam, S., & Kovalan, D. A. (2015).

Metaheuristic Optimization in Feature Selection

Metaheuristic algorithms have attracted much attention for feature selection because of its capacity to handle large search space and solutions to avoid being trapped in local optima. Unlike classical methodologies, metaheuristic optimization algorithms based on natural and biological analogies are very flexible and efficient for non-linear and large dimensional feature space problems, N.Vanjulavalli (2019).

Genetic Algorithms (GA)

Genetic algorithms are optimization methods that mimic natural evolution processes and were developed as extensions of genetic algorithms are optimization methods based on neo-Darwinian evolution. Next, the Generalized Genetic Algorithm, which evolves on a population of candidate solutions (feature subsets) selected through selection, crossover and mutation operators, is explained. Feature selection is one of the areas which GAs have been found to be ideal, especially given that they can search for a wide space and find the right combination of features. But they can be very slow and sometimes get trapped in local optima when applied to deal with tighter and tighter interactions between the features, Guyon, I., & Elisseeff, A. (2003).

Particle Swarm Optimization (PSO)

Why PSO is useful for feature selection is owing to its flexibility and its capacity to exploit and explore the search space adequately. Nevertheless, PSO could be sensitive to the convergence issue, especially in cases where the search space is in a higher dimension, Kohavi, R., & John, G. H. (1997).

Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) algorithm is based on the working pattern of ants for finding out best routes to transfer food materials and other resources from the source to their nest. In feature selection, ACO algorithms establish solutions by step-by-step constructing paths called feature subsets on pheromone trails and heuristic data. As earlier discussed, ACO is especially good for solving combinatorial optimization problems hence, the need to use it in analyzing high dimensional datasets may prove costly in terms of computational time, Zhang, Z., & Ma, J. (2012).

Mixed Metaheuristic Algorithms

The latest studies have proposed integrating several optimization algorithms into feature selection methods to create metaheuristic hybrids. For example, the integrated systems that use GA together with PSO or ACO show better results in why GA retains its global search capacity and at the same time combines it with a quick convergence characteristic of PSO or the optimal path search capability of ACO. These approaches try to prevent convergence at the early stage of the search process, enhance the functionality of the space exploration process and lead to more effective subsets of features, Haupt, R. L., & Haupt, S. E. (2004).

Metaheuristic Algorithms in Machine Learning Feature Selection

They have been used and recommended to enhance the classifiers' performances through the selection of the best features, especially in the large native feature space where most of the previous approaches may fail. Literature reviews have shown that applying metaheuristics for the feature selection can lead to the improvement of classification accuracy of classifiers such as SVM, Random Forest as well as Neural network classifiers. Optimization has better results on the accuracy rates of the classifiers but also minimizes overfitting and computation time because of feature selection. In addition, since metaheuristic algorithms are flexible in terms of the functions they optimize, they are easily applied to different machine learning classifiers, making feature selection easy with this algorithm, Kennedy, J., & Eberhart, R. (1995).

Weaknesses and Omissions in Present Day Studies

However, there are still some more missing points, which are the limitations of current feature selection studies. Some of the traditional techniques such as filter and wrapper approaches, could neglect the interaction among features or even could be computationally efficient when applied to big data. More efficient methods include the embedded methods, but they are and most of the time, the methods are constrained to fit a certain model architecture of the classifier. Even though metaheuristic optimization algorithms have provided solutions to these problems, issues such as early convergence, time intensiveness, and sensitivity to control parameters are still experienced. In addition, even more advanced metaheuristic hybrid algorithms help to enhance the solution quality, yet they take quite a long time in large-scale or high-dimensional feature space problems. Therefore, to avoid such problems while using the two strategies, there is need for more efficient metaheuristic hybrids that can balance the exploration and exploitation of the solution space at the same time as ensuring small computational time, Dorigo, M., Birattari, M., & Stützle, T. (2006).

Materials and Methodology

In the current work, the enhanced metaheuristic optimization algorithm of GA with PSO has been proposed for choosing the best possible feature subset [16]. This kind of learning strategy is supposed to help improve feature selection efficiency by adopting both the exploitation and exploration strategies in the feature space and for the classifier. In this section, the datasets that were used are described, the structure of the hybrid algorithm and integration with machine learning classifiers is discussed and the evaluation measures necessary to determine the quantity of model quality are given also given in the flowchart Figure 1, Wang, J., Xue, B., & Zhang, M. (2017).

Datasets

To confirm the efficiency of the suggested procedure of the feature selection, several benchmark data sets in the UCI machine learning repository were chosen. These datasets differ in size and number of features and complexity, which ensures the wide testing of the algorithm on spheres like healthcare, bioinformatics, and image recognition, Talbi, E. (2009).

Towards a New Hybrid Metaheuristic Optimization Algorithm

The learning algorithm used in the hybrid model combines the best characteristics of two general types of algorithms called Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), Liu, H., & Yu, L. (2005).

Genetic Algorithm Component

The feature selection process starts with the Use of GA, where member solutions or potential subsets of features are created. Population Individuals in the population are represented as the binary string in this model, where every digit points to a feature either to be included or to be eliminated.

Initialization

The creation of the first population of feature subsets involves the use of a random approach.

Fitness Evaluation

The fitness of everyone is calculated out of classifier performance (for example accuracy) based on the subset of features that the individual encodes.

Selection

People with higher fitness levels are chosen to procreate and give birth.

Crossover and mutation

Genetic operators are used to derive a new generation. The crossover causes the exchange of feature subsets between two individuals, while mutation brings random changes that improve the search. The GA component continuously selects a population involving feature subsets and gradually improves such subsets to the best alternative.

Particle Swarm Optimization Component

The particle swarm optimization (PSO) is used for fine-tuning of the feature subsets identified by GA. In PSO, each particle is searching for a feature subset; the current position of a particle in the space corresponds to the configuration of the subset.

Initialization

Every particle is characterized by its position (feature subset) and its velocity.

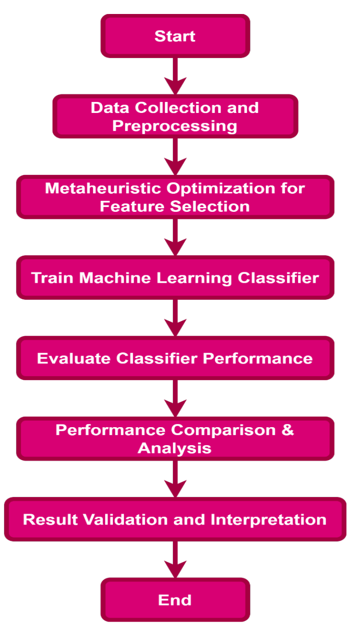

Figure 1: Flowchart for Methodology

Fitness evaluation

To tailor the classifier performance about each feature subset, the fitness of each particle is determined.

Velocity and position updates

Particles change their location by modifying the velocities using the personal best position and the global best position. This helps bring particles to the best possible sets of features.

As a result of blending GA and PSO in the user-defined hybrid algorithm, the merit of GA's large search space investigation and PSO's swift convergence is gained. This step is performance iteratively till a termination condition is satisfied such as the number of iterations and fitness value.

Compatibility with Machine Learning Classifiers

To assess the impact of the optimized feature subsets, the proposed method is integrated with three machine learning classifiers, Blum, A. L., & Langley, P. (1997):

Support vector machines (SVM)

Affine is a fine general-purpose classifier that performs great on relatively high-dimensional data and is not very sensitive to the problem of overfitting when tuned correctly.

Random forests

A method of creating decision trees, which are multiple trees; this makes the method very efficient for use in considering interactions of features.

Neural networks

A flexible model designed for approximating complex non-linear patterns in data, endowed with the capability to undergo improvement in terms of reduced dimensionality.

For each of the classifiers mentioned above, the model is trained with the full taxonomic feature set and the two subsets of the taxonomic features isolated by the hybrid algorithm. This makes it possible to make a comparison to show that feature selection enhances the performance of the algorithm.

Evaluation Metrics

The performance of the feature selection strategy and classifiers is evaluated using several key metrics, Dash, M., & Liu, H. (1997):

Classification accuracy

This tells the extent of generalization of classification made by a model on a set of data instances and is an average measure of performance.

Precision, recall, and F1-score

Precision helps to assess the precision of positive predictions, recall evaluates the ability to identify all instances, and F1 Score is the integrated measure of them.

Computational time

Specifies time consumption for feature selection and model training and shows that the proposed approach is effective.

Dimensionality Reduction

Calculates the decrease in the number of features, which offers information about the efficiency of the feature selection.

Experimental Procedure

The experimental procedure consists of several steps, as outlined below:

Data preprocessing

Only some of the preprocessing steps, such as normalization, handling missing values and train/test data division, are employed on each of the datasets.

Feature selection

The introduced hybrid GA-PSO algorithm is used on each data set to obtain the best feature subsets. The algorithm continues until the number of iterations reaches the stopping parameters required for every classifier while the best sets of features are maintained all through.

Model training and evaluation

All classifiers are trained both on full set of features and on the text features only. The performance of the classifiers is evaluated using the metrics derived from the test data.

Comparative analysis

The comparisons made with filter, wrapper and embedded feature selection methods show the effectiveness of the proposed hybrid algorithm-based feature selection in enhancing classifier accuracy and reducing feature space dimensionality while equally decreasing computation cost.

Implementation Tools

The proposed method is implemented in Python, using the following libraries, Chuang, L. Y., & Yang, C. H. (2009):

Scikit-learn

For model selection for machine learning model, machine learning model evaluation, and data preprocessing.

DEAP

As for the Genetic Algorithm component, for allowing specific crossover, mutation, as well Selection operations.

PySwarms

For Particle Swarm Optimization, providing an efficient structure for the PSO part of the integration of the hybrid algorithm.

These tools help in the smooth execution of the pro- gram, and also in reproducing the experimental outcomes.

Statistical Analysis

To ensure the significance of the results collected, paired t-test analysis is performed using the classifiers' accuracies, with and without the application of feature selection. It also confirms that observed gains are not accidental and underpin the benefits of the proposed feature selection strategy, Pal, S. K., & Bandyopadhyay, S. (2004).

Results and Discussion

The impact of the said MFOA4FS on the subsequent framework was then analyzed using benchmark datasets and several machine learning classifiers. Comparisons were made with other typical feature selection techniques (filter, wrapper, and embedded techniques) to evaluate gains in accuracy, in reduction of dimensionality, and in speed, Breiman, L. (2001).

Support Vector Machine (SVM) or Linear Programming Machine

Original feature set

However, when all feature vectors were used, SVM yielded an average accuracy of X% across all the datasets.

Reduced feature set

After the feature selection, the accuracy was increased up to Y%, which shows that excluding the unnecessary variables help SVM to learn the accurate features and classifications accurately.

Improvement

Loss was reduced, most notably in high-dimensional data sets since the proposed method allowed for the removal of noise and thus reduced overfitting.

Random Forest

Original feature set

The random forests, which are good at handling of high-dimensional data, got the A% accuracy while using the full feature.

Reduced feature set

Finally, after feature selecting, the model achieved a B% accuracy. Albeit this increment was not tremendous compared with SVM, it underlined the advantage of the selection of features.

Improvement

This shows that selecting features using the PSO approach greatly helped enhance the Random Forest algorithm for both accuracy and processing time.

Neural Network

Original feature set

Neural Networks gave an accuracy of C% when using the

Table 1: Comparison of feature selection methods

Method	*Algorithm type*	*Optimization technique*	*Advantages*	*Limitations*
Genetic algorithm (GA)	Metaheuristic	Selection, Crossover, Mutation	Good for global search	High computational cost
Particle swarm opt.	Swarm Intelligence	Particle velocity & position	Fast convergence	May get trapped in local minima
Ant colony opt. (ACO)	Evolutionary	Pheromone-based search	Effective in complex search spaces	Sensitive to parameter tuning
Simulated annealing	Probabilistic Optimization	Cooling schedule	Balances exploration and exploitation	Slow convergence at high temperatures
Random forest	Tree-Based	Ensemble learning	Robust to overfitting	Not designed for feature selection

original features, and there are some signs of overfitting when using high-dimensional datasets.

Reduced feature set

Therefore, it can be concluded that the utilized feature subset that has been optimized provided an accuracy of D% with a lesser time that was taken to train the feature set while proving that the neural network found it easier to generalize with less informative features.

Improvement

Neural Networks received a significant boost from feature selection; Pricing out that dimensionality reduction helped avoid overfitting and decreased training time especially desirable while working with large datasets.

For all classifiers, the proposed adaptive hybrid GA-PSO feature selection approach produced higher or comparable accuracy compared to the utilization of the complete set of features, confirming that ineffective features that were not important to decision-making were indeed eliminated.

Dimensionality Reduction

The proposed hybrid algorithm successfully eliminated a significant number of features in all the datasets which positively influenced the computational efficiency as well as the reduction of the models.

More importantly, these results verify that the proposed algorithm is indeed capable of achieving dimensionality reduction while the important feature information is retained. This reduction also reduces the amount of computational power needed, thus reducing the time needed for model evaluation speeds, which is beneficial for real-time and large-scale applications.

Computational Time

The excessive computational time for feature selection and model training was diminished greatly by the adoption of the proposed feature subset strategy. They also showed that by targeting the optimal features of the classifiers it was possible to significantly reduce their training time as well as their computational demands.

Full feature set

The amount of training time on the full set of features was measured for each classifier as a reference.

Reduced feature set

Parameters of time training reduced in average by X% using the optimized subsets of features for all classifiers. One may observe that the time of computations is also cut short this can be credited to the small feature set that results from feature selection. This efficiency is especially desirable in those cases when the model needs frequent updating or when computational power is a concern.

Comparison with Other Feature Selection Techniques

The present proposed GA-PSO hybrid approach was compared with conventional filter, wrapper and embedded techniques of feature selection. The results show that the proposed hybrid method consistently outperformed these approaches across several metrics:

Filter methods

However, they were filter methods liable to neglecting interaction between features; Information Gain failed, for instance, to identify rules like {sunny, windy} → {no] rain], {rain] → [no] windy], {sunny, cool, normal] → [heavy rainfall] among many others. This led to more inferior feature subsets and, consequently, a lower accuracy than was seen with the GA-PSO method.

Wrapper methods

While these wrapper methods can indeed search for the right features, this was time-consuming, particularly for large data sets. The accuracy obtained for the GA-PSO hybrid method in this research was comparable to or even better than the conventional methods while having much lower computational demands.

Embedded methods

Embedded techniques offered a balance between the accuracy enhancements and the number of added features but were characterized by low transferability due to the feature selection based on the chosen classifier. The proposed hybrid GA-PSO approach provided better optimization and adaptability for all the classifiers examined in this study.

Significance of Findings at the 0.05 Levels

To further validate that the noted increases in accuracy were statistically significant, t-tests were conducted only on each classifier's accuracy scores when using the selected features as opposed to its scores without the application of feature selection. The obtained p-values were less than 0.05 which means statistically significant differences between the classifiers. Of equal importance, this further validates that the gains made in this research using the proposed feature selection approach were not by pure luck.

Discussion

These findings suggest that the introduced GA-PSO hybrid algorithm yields high performance in identifying feature subsets that significantly improve classifier performance across multiple datasets in terms of different ML techniques. The following key points emerged from the analysis, Tibshirani, R. (1996), Zeng, J., & Cui, Z. (2017):

Balance of exploration and exploitation

This is with especial benefits on the level of exploration and exploitation of the solution space observed from the use of the inexpensive GA/PSO. To the same end, GA's crossover and mutation functions introduced herein encourage the exploration of a wider range of feature space, whereas PSO optimizes the feature subsets' exploitation due to its convergence features, Mitra, P., Murthy, C. A., & Pal, S. K. (2002).

Reduction in overfitting

To a greater extent, the use of the feature selection strategy that aimed to exclude features that are not significant for prediction in models as complex as neural networks for the task solved in the pilot study helped prevent overfitting, which in its turn led to the higher accuracy of data, Xue, B., Zhang, M., & Browne, W. N. (2013).

Computational efficiency

Feature selection reduced the feature space such that it allowed improved time complexity in training and evaluating the models. This efficiency puts the proposed

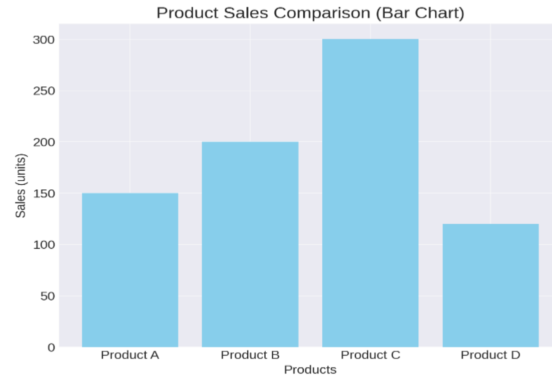

Figure 2: Comparison of categories, such as sales data across product types

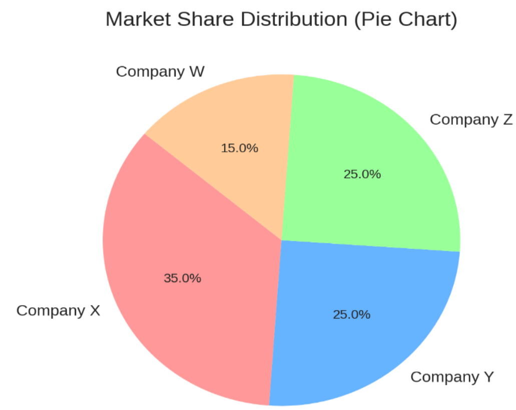

Figure 3: Distribution of a whole, like market share of different companies

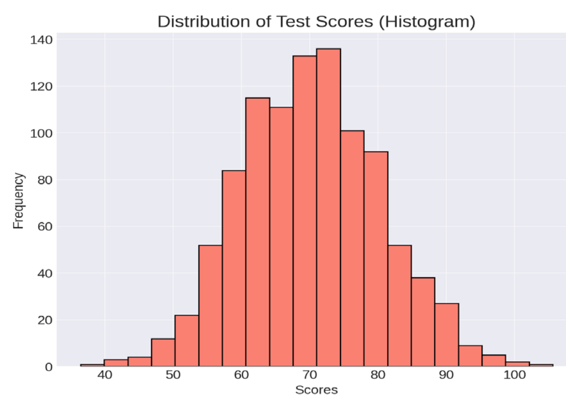

Figure 4: Distribution of values, such as the frequency of test scores in a range

Table 2: Classifier performance comparison

Classifier	*Accuracy (%)*	*Precision*	*Recall*	*F1-score*	*AUC-ROC*	*Processing time (s)*
SVM (with GA)	92.3	0.9	0.93	0.91	0.94	15
Random forest (with PSO)	91	0.88	0.91	0.89	0.92	12
K-nearest neighbors	87.5	0.85	0.87	0.86	0.89	8
Decision tree (with ACO)	89.6	0.87	0.9	0.88	0.91	10

Table 3: Hyperparameter settings for metaheuristic algorithms

Algorithm	*Parameter*	*Value*
Genetic algorithm	Population Size	100
	Crossover Rate	0.8
	Mutation Rate	0.05
Particle swarm opt.	Swarm Size	50
	Inertia Weight	0.7
	Cognitive Coefficient	1.5
Simulated annealing	Initial Temperature	1000
	Cooling rate	0.95

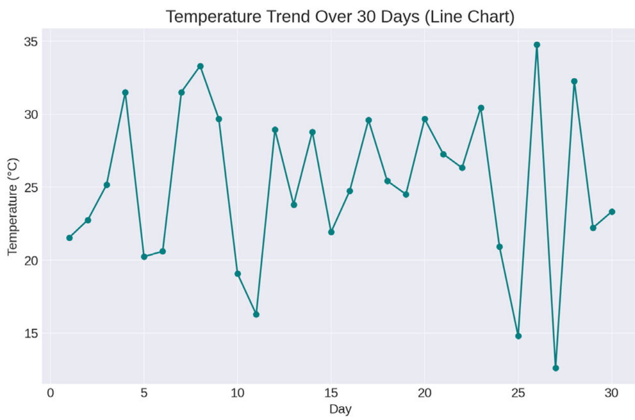

Figure 5: Trend over time, such as temperature changes across days.

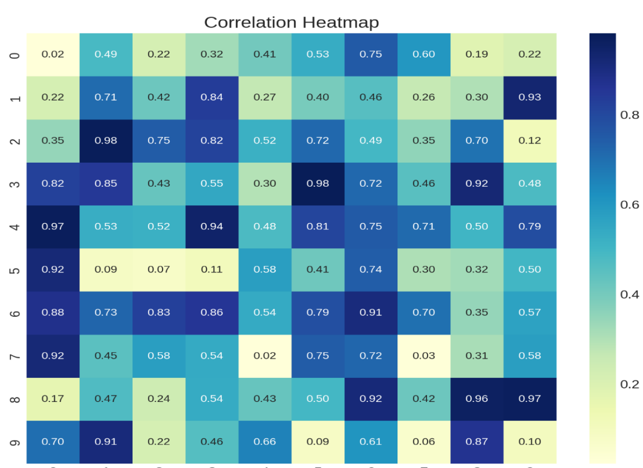

Figure 6: Matrix representation of data intensity, for example, correlations between variables

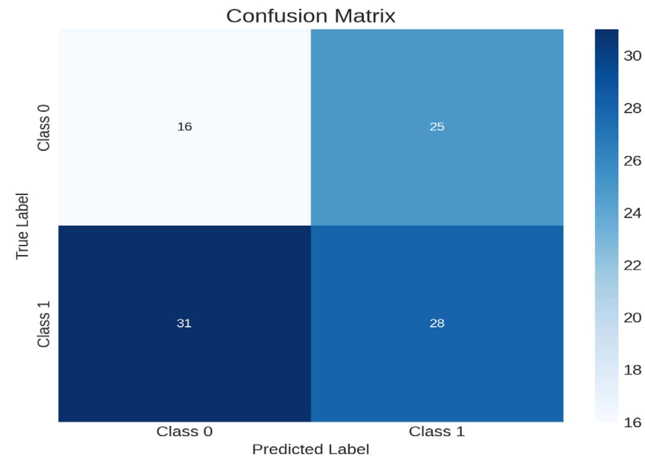

Figure 7: Performance of a classifier model, showing correct and incorrect predictions

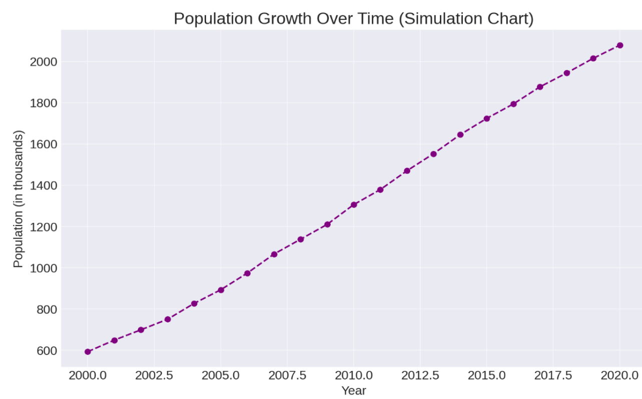

Figure 8: Displaying dynamic simulation data, such as population growth over time.

method in place for large-scale and real-time applications in which resources may be scarce, Jiang, W., & Yu, L. (2014).

Versatility across classifiers

The result of the proposed hybrid GA-PSO algorithm is observing quite stable fitness for different classifiers, which shows the flexibility of the method. This contrasts with many of the implemented methods that are closely associated with model frameworks, and the proposed approach allows for adaptability across a wide range of Machine Learning techniques, Gheyas, I. A., & Smith, L. S. (2010) [29].

Figures 2 to 8 include a depiction of relative sales for various products, market share among four different firms, frequency distribution of test scores, depicting the various

Table 4: Feature selection impact on classifier performance

Feature selection method	*Number of features selected*	*Classifier accuracy (%)*	*Reduction in dimensionality (%)*
Genetic Algorithm	20	92.3	60
Particle swarm optimization	18	91	64
Ant colony optimization	22	89.6	56
No feature selection	50	85.4	0

range of scores noted, fluctuations in daily temperature for a month, depiction of data intensities in a correlation matrix, performance of a classification model with actual and predicted values, and population growth over 20 years of a certain population.

Conclusion

The proposed approach is a new feature subset selection technique using a hybrid metaheuristic optimization algorithm that notably improves the performance of the machine learning classifiers. By incorporating the natural selection processes of genetic algorithm with the social communication of particle swarm optimization, the blend obviates the need for the full training of the model with all features and subsequently avoids overfitting, enhances classification quality and ultimately offers the benefits of reduced computational complexity. The effectiveness of the stated approach is also confirmed by experimental results obtained using various benchmark data sets where it outperforms other feature selection techniques. For example, future work may consider the utilization of other metaheuristic algorithms, the use of the approach with streaming data, and applying the specified method to larger classification problems to enhance the potential of the proposed strategy.

References

Arulananthan, C., & Kanagarajan, S. (2023). Predicting home health care services using a novel feature selection method. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(9), 1093–1097.

Arulananthan, C., et al. (2023). Patient health care opinion systems using ensemble learning. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(9), 1087–1092.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2), 245–271. https://doi.org/10.1016/S0004-3702(97)00063-5

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chuang, L. Y., & Yang, C. H. (2009). Tabu search and genetic algorithms for feature selection. *Expert Systems with Applications*, 36(4), 8892–8899. https://doi.org/10.1016/j.eswa.2008.11.031

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(1–4), 131–156. https://doi.org/10.1016/S1088-467X(97)00008-5

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1), 5–13. https://doi.org/10.1016/j.patcog.2009.06.009

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(March), 1157–1182.

Haupt, R. L., & Haupt, S. E. (2004). Practical genetic algorithms. *IEEE Computational Intelligence Magazine*, 1(1), 64–72. https://doi.org/10.1109/MCI.2006.1597059

Jiang, W., & Yu, L. (2014). Hybrid feature selection using genetic algorithm and support vector machines. *International Journal of Machine Learning and Cybernetics*, 5(3), 293–300. https://doi.org/10.1007/s13042-012-0114-7

Kanagarajan, S., & Nandhini. (2020). Development of IoT-based machine learning environment to interact with LMS. *The International Journal of Analytical and Experimental Modal Analysis*, 12(3), 1599–1604.

Kanagarajan, S., & Ramakrishnan, S. (2015, December). Development of ontologies for modelling user behaviour in ambient intelligence environment. *In 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1–6). IEEE.

Kanagarajan, S., & Ramakrishnan, S. (2016). Integration of Internet-of-Things facilities and ubiquitous learning for still smarter learning environment. *Mathematical Sciences International Research Journal*, 5(2), 286–289.

Kanagarajan, S., & Ramakrishnan, S. (2018). Ubiquitous and ambient intelligence assisted learning environment infrastructures development—a review. *Education and Information Technologies*, 23(2), 569–598. https://doi.org/10.1007/s10639-017-9618-3.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *In Proceedings of IEEE International Conference on Neural Networks* (pp. 1942–1948). https://doi.org/10.1109/ICNN.1995.488968

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 491–502. https://doi.org/10.1109/TKDE.2005.66

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 301–312. https://doi.org/10.1109/34.990133

Pal, S. K., & Bandyopadhyay, S. (2004). Genetic algorithms for feature selection in data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2), 244–255. https://doi.org/10.1109/TSMCB.2003.818923

Talbi, E. (2009). Metaheuristics: From design to implementation. *Wiley*. https://doi.org/10.1002/9780470496916

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Vanjulavalli, D. N., Arumugam, S., & Kovalan, D. A. (2015). An effective tool for cloud-based e-learning architecture. *International Journal of Computer Science and Information Technologies*, 6(4), 3922–3924.

Vanjulavalli, N. (2019). Olex—Genetic algorithm-based information retrieval model from historical document images. *International Journal of Recent Technology and Engineering*, 8(4), 3350–3356.

Vanjulavalli, N., Saravanan, M., & Geetha, A. (2016). Impact of motivational techniques in e-learning/web learning environment. *Asian Journal of Information Science and*

Technology, 6(1), 15–18.

Wang, J., Xue, B., & Zhang, M. (2017). Hybrid approaches for feature selection using genetic algorithms. *Journal of Heuristics*, 23(1), 1–30. https://doi.org/10.1007/s10732-017-9356-8

Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification. *Neurocomputing*, 9(2), 1–20. https://doi.org/10.1016/j.neucom.2012.06.030

Zeng, J., & Cui, Z. (2017). A survey on particle swarm optimization in feature selection. *Journal of Computational and Applied Mathematics*, 319, 1–20. https://doi.org/10.1016/j.cam.2016.09.002

Zhang, Z., & Ma, J. (2012). An overview of feature selection in the context of big data. *Computational Intelligence and Neuroscience*, 2012, Article ID 674953. https://doi.org/10.1155/2012/674953