Scientific Journal of Informatics
Vol. 13, No. 1, February 2026
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Gene Expression-Based Lung Cancer Prediction in Smokers Using SVM and Moth-Flame Optimization Algorithm

Salma Safira Ramandha[1], Angel Metanosa Afinda[2], Isman Kurniawan[3]
[1,3]School of Computing, Telkom University, Bandung, Indonesia
[2]School of Electrical Engineering, Telkom University, Bandung, Indonesia

Abstract.
Purpose: Predicting lung cancer from gene expression data is challenging. These datasets usually contain tens of thousands of genes, and only a small fraction of them truly help the classifier; many others add unnecessary variation and make the model more sensitive to noise. The problem becomes even more complicated when the samples come from smokers, whose gene expression patterns tend to be more irregular. Considering these challenges, this study aims to develop a more dependable prediction model by first reducing the feature using the Moth-Flame Optimization (MFO) algorithm and then training a Support Vector Machine (SVM) with tuned hyperparameters. This combination is expected to yield a model that performs more consistently and achieves better accuracy than a baseline SVM trained without feature selection.
Methods: This study uses GSE4115 dataset, which contains 22,215 gene expression features. MFO is used to select the most informative subsets, and hyperparameter tuning is applied to optimize SVM across multiple kernels. The workflow begins with data cleaning and normalization, followed by identifying the most suitable feature subset using the MFO algorithm. After the features are narrowed down, the SVM model is tuned through hyperparameter optimization, and the final stage evaluates how well the model performs.
Results: MFO algorithm was used to select the best feature from the gene dataset, leaving only a small set of genes that contributed the most to the classification. Using this subset feature, the SVM model's kernel showed solid performance. On the test data, polynomial kernel using 286 MFO-selected features reached 0.84 accuracy and an f1-score of 0.85. These findings indicate that the model can distinguish between cancer and non-cancer cases in smokers with strong level of accuracy. The use of MFO alongside SVM also contributes to reducing much of the noise that typically arises in high-dimensional gene-expression datasets, allowing the classifier to focus on more meaningful patterns.
Novelty: This study brings a different angle by pairing the MFO method with an SVM classifier for predicting lung cancer in smokers. This combination has not been used in gene-expression studies. The idea is to cut down the large number of genes into a smaller set before building the classifier. With this setup, the model becomes simpler to train and able to deliver better prediction results, offering a practical way to improve early detection.

Keywords: Lung cancer, Gene expression, Support vector machine, Moth flame optimization, Metaheuristics
Received December 2025 / **Revised** January 2026 / **Accepted** January 2026

INTRODUCTION

Lung cancer is one of the leading causes of cancer-related death worldwide [1]. In many countries, including Indonesia, the burden of lung cancer continues to increase and is closely associated with smoking and exposure to harmful pollutants rising from 1.4 per 1,000 individuals in 2013 to 1.8 per 1,000 individuals in 2018 [2], [3], [10]. Patients are often diagnosed at an advanced stage, when symptoms such as persistent cough, hemoptysis, chest pain, and recurrent infections have already appeared, and treatment options have become more limited [4]. Conventional imaging techniques, such as chest radiography and Computed Tomography (CT), are widely used in clinical settings, but they still struggle to detect early-stage lesions and may produce false positives if used without careful clinical evaluation [6], [10].

Improvements in computational methods have led to new ways to diagnose diseases using machine learning and gene expression analysis. Gene expression data can detect molecular-level changes prior to the visual identification of tumors, rendering it a promising method for early cancer prediction [7]. Machine learning

[1]*Corresponding author.
Email addresses: salmasafira@student.telkomuniversity.ac.id (Ramandha), angelmetanosa@telkomuniversity.ac.id (Metanosa), ismankurniawan@telkomuniversity.ac.id (Kurniawan)
DOI: 10.15294/sji.v13i1.38268

techniques have shown great promise in using gene expression profiles to classify lung cancer. In fact, they have done better than traditional imaging in several studies [4], [12].

To overcome these limitations, gene expression profiling has been introduced as an alternative approach for early cancer detection because it can capture molecular changes before structural abnormalities become visible [5] ,[7]. With the development of microarray and high-throughput technologies, many studies have applied machine learning methods to classify lung cancer and related diseases based on gene expression profiles [8], [11], [22], [32]. Classifiers such as Support Vector Machine (SVM), Random Forest, ensemble methods, and deep learning architectures have shown strong performance on various gene expression datasets [35]. However, gene expression datasets typically contain tens of thousands of genes but only a limited number of samples, which creates a high-dimensional setting that easily leads to overfitting and unstable models if feature selection is not handled properly [18], [15], [17], [20].

Previous research on lung cancer gene expression has mostly used other optimization or feature selection methods, such as Particle Swarm Optimization, Genetic Algorithms, or filter and wrapper techniques, often combined with a single SVM kernel or other classifiers [11], [34], [35]. Studies that specifically analyze gene expression in smokers and use airway epithelial samples, such as those similar to the GSE4115 setting, are still relatively few [32]. Furthermore, although SVM is widely recognized as a strong baseline for high-dimensional classification, most works do not systematically compare different SVM kernels on the same optimized feature subsets produced by a metaheuristic algorithm [14], [19], [21], [31]. This leaves an open question about how optimization-based feature selection interacts with kernel choice and how this combination affects accuracy, stability, and generalization in the specific case of gene expression data from smokers.

Despite these promising developments, multidimensional gene expression data presents significant challenges [25], [27]. Excessive features can degrade model performance without effective feature selection [24], [31]. Metaheuristic optimization techniques have therefore gained interest, with Moth-Flame Optimization (MFO) demonstrating strong capability in adaptively exploring the search space. MFO is one such technique that is gaining attention for its ability to adaptively search for the optimal feature subset using swarm intelligence, thereby reducing the feature count without sacrificing critical information [16], [29]. From the literature we reviewed and the initial analysis we conducted, a clear research gap emerged. To our knowledge, no prior work has applied MFO specifically for feature selection in combination with SVM to classify lung cancer using gene-expression data from smoker-based airway samples such as GSE4115 [14], [36].

This research aims to develop an accurate and efficient lung cancer prediction model for high-risk smokers using gene expression data. The proposed method integrates the MFO algorithm for feature selection with the robust SVM classifier, leveraging MFO's global search ability to identify the most relevant gene features.

METHODS

In this study, the lung cancer prediction process using gene-expression data was carried out through several structured steps. These include data preprocessing, splitting the dataset, selecting features with the MFO algorithm, tuning the SVM hyperparameters, building the classification model, and finally evaluating its performance. The overall research workflow is illustrated in Figure 1.

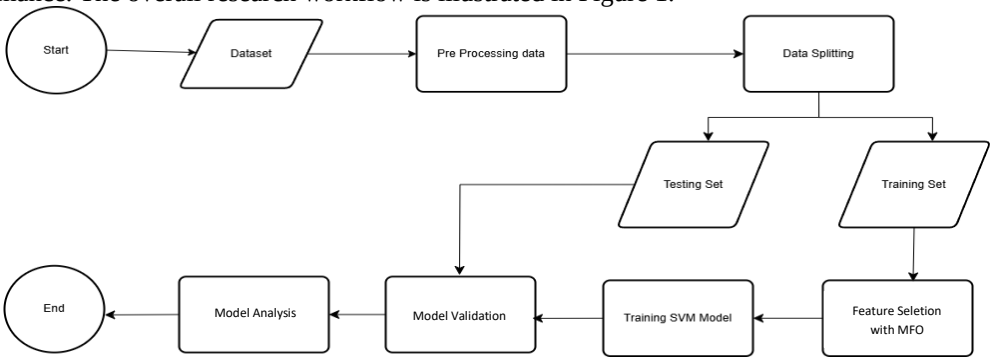


Figure 1. Research methodology flowchart

A. Dataset

Our study leverages the GSE4115 dataset from the Gene Expression Omnibus (GEO), which contains 22,215 gene expression features measured from bronchial epithelial samples. Although this dataset has been used in several previous lung cancer classification studies, the raw labels needed to be adapted to match our binary prediction goal [5],[8]. This dataset utilizes microarray technology, which is a high-throughput method employing microscopic DNA spots on a solid surface to simultaneously analyze the expression levels of thousands of genes [28], [30]. The original three categories (diagnosed, not diagnosed, suspected) were simplified by removing the suspected class entirely. The resulting dataset contains 187 samples, consisting of 97 diagnosed and 90 not diagnosed cases, which is sufficiently balanced to reduce the risk of class bias. For model validation, we applied a 70:30 train–test split, as summarized in Figure 2.

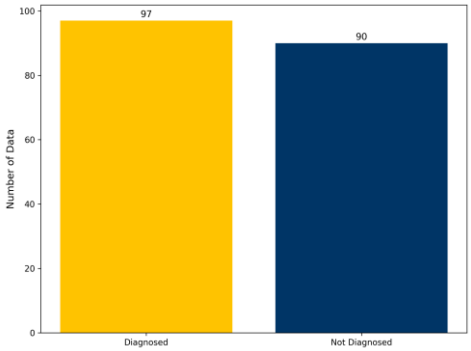


Figure 2. Number of samples in dataset

B. Moth Flame Optimization

The Moth-Flame Optimization (MFO) algorithm is a nature-inspired optimization method that has attracted considerable attention because it is simple and performs well on various complex problems MFO is easy to implement, requires only a few control parameters, and does not depend on gradient or derivative information [9], [13]. It mirrors how moths move around a light source in a spiral formation [13]. In feature selection, each candidate solution, identified as a moth, is a binary vector that encodes the selected features in the search space. The best solutions found during the iterative process are like flames that lead the moths to promising areas [29], [40]. This study's fitness function is defined as equation (1):

$$f(x) = a \times (1 - P) + (1 - a)x \frac{N_{selected}}{N_{features}} \quad (1)$$

To better understand the multi-objective optimization problem, it's needed to define the variables that the feature selection fitness function uses. P is the classifier's classification accuracy on the evaluated feature subset, $N_{selected}$ is the number of features that the MFO agent picked as selected, and $N_{features}$ is the total number of features in the original dataset. Finally, a is the weighting parameter is an important part that is usually set between 0 and 1. It controls how crucial it must be to optimize classification accuracy than to minimize the size of the final feature subset [6], [29]. Figure 3 shows the flowchart for the algorithm used in this study.

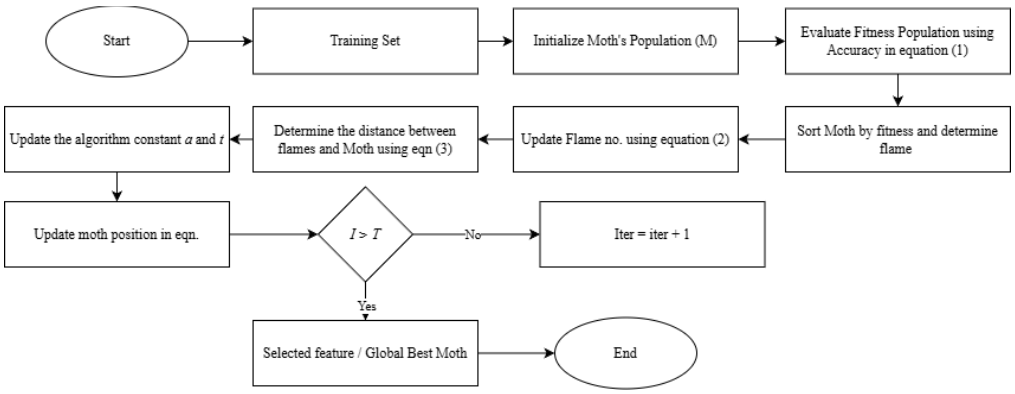


Figure 3. MFO based feature selection in the proposed model

The formula in equation (2) says that the algorithm changes the direction of the search during the iterative process, and the number of active flames goes down in a straight line over time.

$$FlameNo = round(N - l * \frac{N-1}{T}) \tag{2}$$

Equation (3) says that the moth moves toward its corresponding flame at each step based on their distance D_i. This makes sure that the change from exploration to exploitation goes smoothly.

$$D_i = |F_j - M_j| \tag{3}$$

Equations (4) and (5) are then used by the MFO to update the control constants a and t, where t is a random number in the interval [-1, 1]:

$$a = -1 + l\ x\ \left(\frac{-1}{T}\right) \tag{4}$$

$$t = (a-1)\ x\ rand + 1 \tag{5}$$

The spiral function in equation (6), which mathematically abstracts the moth's navigation path around a light source, is then used to update the moth's position. When the maximum number of iterations T is reached and the overall best-found solution is obtained, the process comes to an end [7].

$$S(M_i, F_j) = D_i . e^{bt} . \cos 2\pi t + F_j \tag{6}$$

The MFO implementation used a set of basic parameters to make sure that the results were the same every time and could be repeated. A seed value of 42 and a maximum number of iterations of 120 were set. Also, the weighting parameter, alpha, was set to 0.90. Table 1 gives a brief overview of all the factors that were looked at in this study.

Table 1. Parameter used in the MFO algorithm

Parameter	Values
max_iters	120
alpha	0.90
seeds	42
population_size	[80,100]

We varied the size of the population to see how optimization worked, which led to six different experimental schemes, as shown in Table 2. These schemes put run into groups of 80 and 100 population size across linear, RBF, and polynomial kernel.

Table 2. Experimental schemes based on kernel type and MFO population size

Scheme	SVM Kernel	MFO Population Size
LNR 80	Linear	80
LNR 100		100
RBF 80	RBF	80
RBF 100		100
POLY 80	Polynomial	80
POLY 100		100

C. Support Vector Machine

A supervised learning algorithm called Support Vector Machine (SVM) sorts data by finding the best boundary that separates one class from another [23], [31]. SVM does not only rely on a simple linear split; it maps the input data into a higher dimensional space and looks for a separating hyperplane that provides the widest possible margin between classes [23]. The margin is the space between the hyperplane and the closest training samples, which are called support vectors. In general, a wider margin means that the model will work better on new data [26]. Figure 4 shows the separating hyperplanes used in this study.

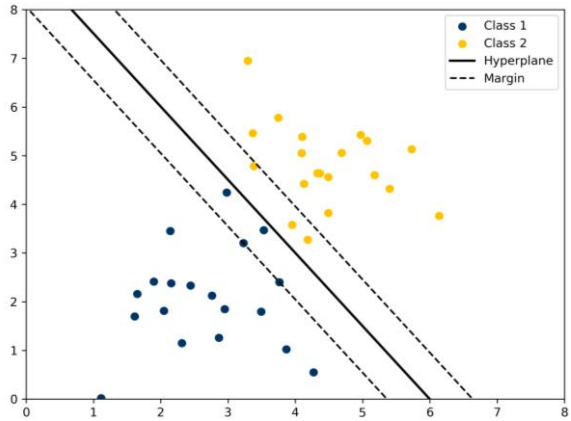


Figure 4. Hyperplane illustration of SVM

For linearly separable data, the hyperplane's classification boundary is defined by the following equation (7):

$$f(x) = w \cdot x + b \tag{7}$$

where *x* is the input feature vector, *w* is the weight vector, and bis is the bias term. SVM uses a variety of kernel functions to implicitly map the input data into a multidimensional feature space where linear separation is feasible in order to efficiently handle non-linear classification problems [16]. The Linear Kernel, Polynomial Kernel, and Radial Basis Function (RBF) Kernel are frequently utilized kernel types. Equation (8) is used by the Linear Kernel, which is mainly optimized by adjusting the Cost Parameter (C):

$$K(x_i, x) = X_i^T X \tag{8}$$

The RBF Kernel uses the following equation (9) and requires optimization of both the *Cost* (C) dan *Gamma* (γ) parameter:

$$K(x_i, x) = \exp\left(-\gamma . X_i^T X|^2\right) \tag{9}$$

The Polynomial Kernel in the following equation (10) is governed by the Cost Parameter *Cost* (C) dan *Degree* (d):

$$K(x_i, x) = \left(\gamma . X_i^T X + r\right)^p \tag{10}$$

The model development was implemented using Support Vector Classification (SVC). The selection of kernel functions and parameter tuning was integrated directly with the MFO feature reduction process. After MFO generated a reduced subset of informative genes, these features were used as input for the SVM classifier, enabling the model to learn decision boundaries from a more compact and discriminative data representation [8]. The final hyperparameter ranges are summarized in Table 3.

Table 3. Hyperparameter for Model Development

Hyperparameter	Range
C	[0.001, 0.1, 1, 10, 100, 1000]
Kernel	[linear, rbf dan polynomial]
Degree	[1,2,3,4,5,6]
Gamma	[auto, scale]

D. Model Validation

The performance of the trained models was evaluated using four standard classification metrics: accuracy (Q), precision (PR), recall (RC), and F1-score (F1) [26], [33]. These metrics were computed from the confusion matrix using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as shown in Table 4. Equations (11)–(14) show how each metric is calculated. To assess generalization, we reported the model performance on both the training and testing sets. This comparison helped us detect overfitting and evaluate whether MFO-based feature reduction enabled the SVM classifier to recognize patterns reliably on unseen samples.

Table 4. Confusion matrix

		Predicted Class	
		Negatives	Positives
Actual Class	Negatives	TN	FP
	Positives	FN	TP

$$Q = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$RC = \frac{TP}{(TP + FN)} \tag{12}$$

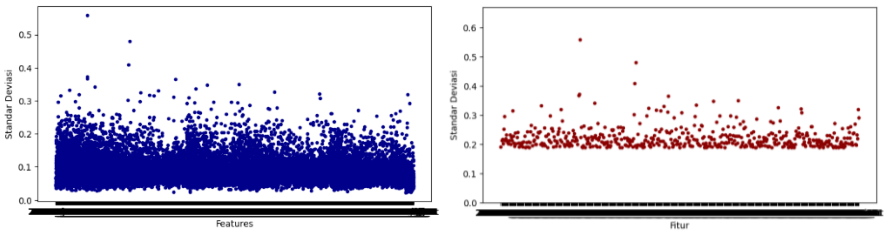
$$PR = \frac{TP}{(TP + FP)} \tag{13}$$

$$F1 = 2\,x\,\frac{PR\,x\,RC}{(PR + RC)} \tag{14}$$

RESULT AND DISCUSSION

A. Feature Selection

Preliminary filtering involved a Variance Threshold of 0.035, condensing the dataset from 22,215 genes to a manageable 519. After this step, MFO was applied to perform feature selection. It is important to clarify that in this study, MFO is used exclusively for selecting the relevant features. MFO does not tune SVM hyperparameters simultaneously; instead, hyperparameter optimization is conducted separately through Grid Search Cross Validation after the feature subset is determined. Ideally, we aimed to do more than just cut features; we analyzed the convergence stability of MFO relative to population size. The convergence plots illustrate the optimization trajectory, where a minimizing score reflects an optimized solution. Figure 5 shows the distribution of standard deviation values before and after variance threshold was applied. The pre-threshold plot displays a dense spread of low variance features clustered near zero, indicating a high proportion of non-informative genes. After filtering, the feature space becomes considerably cleaner, with retained genes exhibiting noticeably higher variance.



(a) (b)

Figure 5. Standard deviation (a) before and (b) after applying variance threshold.

Looking at Figure 6, the comparison between population sizes (80 vs. 100) is revealing. The larger population size resulted in less dropping curves and a more stable objective score. This suggests that expanding the population size enhances the exploration capability, preventing premature convergence.

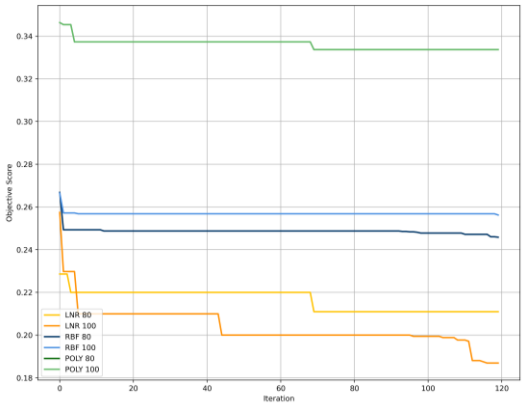


Figure 6. Convergence plot of the MFO-based feature selection with population size of 80 and 100

From these graphs, we saw that the RBF kernel converged the fastest, and the Polynomial kernel followed. The Linear kernel took the most time to optimize. This difference makes sense because non-linear kernels are better at adapting to complicated biological patterns. Linear separation, on the other hand, is more limiting, which makes optimization take longer. Table 5 shows a summary of the selected features and accuracy for each kernel.

Table 5. Model built from the selected features kernel

SVM Kernel	Number of Features	Accuracy
LNR 80	272	0.824
LNR 100	**252**	**0.846**
RBF 80	240	0.778
RBF 100	242	0.767
POLY 80	286	0.690
POLY 100	286	0.690

Comparing the six experimental setups, we noticed that changing the MFO population size affected accuracy in different ways. For the Linear kernel, using a population size of 100 gave slightly better results. The model reached an accuracy of 0.846, a small improvement over the 0.824 recorded with a population of 80. The RBF kernel showed a somewhat similar pattern, but not as clearly. RBF 80 performed better, with an accuracy of 0.778, while RBF 100 dropped to 0.767. This outcome suggests that adding more agents to the population does not automatically improve the search process for this kernel. For the Polynomial kernel, the numbers did not change at all. Both population sizes produced the same accuracy, 0.690, which indicates that the optimization step settled into roughly the same solution regardless of how many candidate agents were involved. LNR 80 and RBF 100 showed the most signs of overfitting. In these cases, training performance was high, but test accuracy dropped a lot. This occurs because smaller or poorly tuned populations may generate feature subsets that detect noise rather than actual biological patterns. Nonlinear kernels involving RBF are especially sensitive to unused noise in datasets with a great deal of dimensions. If we don't remove irrelevant features, these kernels create decision boundaries that are too complicated and fit the training data too closely but don't work well for new data. The Linear kernel, on the other hand, was more stable because it was simpler. This is why LNR 100 always gave the most reliable performance across all MFO population settings. Following MFO selection and subsequent hyperparameter tuning, the performance of these non-linear models surged dramatically, as can be seen in Figure 7.

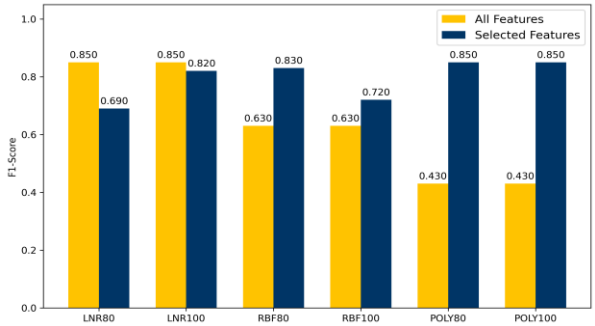


Figure 7. Comparative f1-score all features vs MFO-selected feature subsets.

These results indicate that choosing the correct features is essential for dealing with high-dimensional data. Non-linear kernels tended to overfit if irrelevant genes weren't excluded out. After applying MFO, they resulted in classifiers that were more stable. As a result of this, the Polynomial kernel became a superior model for overall generalization and stability.

B. Hyperparameter Tuning

To be certain the models developed during the feature selection process were powerful, hyperparameter tuning had been performed. We used Grid Search Cross-Validation on each candidate model to determine the best combinations of C, gamma, and degree. Table 6 illustrates a summary of the tuning results.

Table 6. Hyperparameter tuning best result via grid search

Kernel	Best C	Best Gamma	Best Degree
LNR 80	0.1 (1.0)	-	-
LNR 100	0.1 (1.0)	-	-
RBF 80	10 (1.0)	auto (scale)	-
RBF 100	10 (1.0)	auto (scale)	-
POLY 80	1(1.0)	auto (scale)	1(3)
POLY 100	1(1.0)	auto (scale)	1(3)

Figure 8. summarizes the results of hyperparameter tuning, revealing how optimization affects the classification accuracy of models built using MFO-selected feature subsets. The Linear kernel remained the most stable model. Even after hyperparameter tuning, LNR 100 maintained an accuracy of 0.850, which indicated that the suggested by MFO configuration was already reliable. The Polynomial kernel, on the contrary, improved significantly after tuning. After the best parameters were applied, its accuracy increased from 0.690 to 0.790. The RBF kernel was also improved, with both RBF 80 and RBF 100 achieving an accuracy of 0.790. These results indicate that non-linear kernels are additionally adaptable, but their performance depends considerably on selecting the correct hyperparameters.

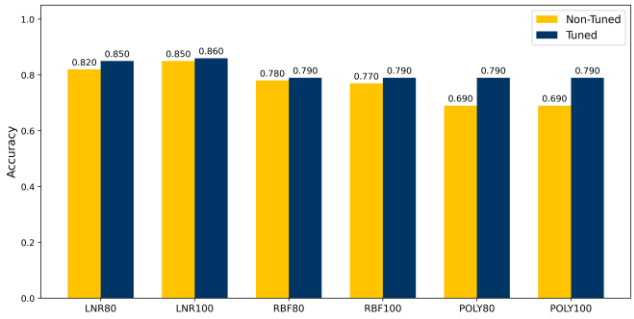


Figure 8. Result of MFO-selected feature subsets on the performance of various tuned SVM kernels

C. Model Validation

The final stage of model validation evaluated the way well each tuned classifier worked on both the testing and training datasets using the features selected by the MFO. Table 7 possesses all the results.

Table 7. Result for subset features train and test data

Model Name	TP	FP	TN	FN	Q	PR	RC	F1
Train								
LNR 80	**61**	**4**	**61**	**4**	**0.94**	**0.94**	**0.94**	**0.94**
LNR 100	61	5	61	4	0.93	0.92	0.94	0.93
RBF 80	59	7	59	6	0.90	0.89	0.91	0.90
RBF 100	61	7	61	4	0.92	0.90	0.94	0.92
POLY 80	63	16	63	2	0.86	0.80	0.97	0.88
POLY 100	63	16	63	2	0.86	0.80	0.97	0.88
Test								
LNR 80	18	6	23	10	0.72	0.75	0.64	0.69
LNR 100	23	5	24	5	0.82	0.82	0.82	0.82
RBF 80	24	6	23	4	0.82	0.80	0.83	0.83
RBF 100	21	9	20	7	0.72	0.70	0.72	0.72
POLY 80	**25**	**6**	**23**	**3**	**0.84**	**0.81**	**0.85**	**0.85**
POLY 100	**25**	**6**	**23**	**3**	**0.84**	**0.81**	**0.85**	**0.85**

The Polynomial kernel performed optimally with both datasets. Although its training accuracy was only 0.86, it showed the smallest generalization gap of 0.02 and highest test performance, with an accuracy of 0.84 and an F1-score of 0.85. It also produced the highest number of true positives (TP = 25) and smallest number of false negatives (FN = 3), indicating that it had greater recall. The linear and RBF kernels performed reasonably well. LNR 100 and RBF 80 both achieved a test accuracy of 0.82, but each had larger generalization gaps of 0.11 and 0.08. On the other hand, LNR 80 and RBF 100 had significant loss in test accuracy, with gaps of 0.22 and 0.20. These results suggest that both models were overfitting on the selected features. overall, the Polynomial kernel proved the most reliable and consistent classifier for the MFO-selected gene subset. The improvements seen in all non-linear models show that MFO successfully removed redundant genes that previously led to overfitting. After selecting features, SVM could learn more useful biological patterns instead of noise. This led to decision boundaries more trustworthy when tested on samples that hadn't been seen beforehand.

CONCLUSION

This study developed a lung cancer prediction model for smokers by using the MFO algorithm together with SVM classifiers. Using MFO allowed us to narrow the original 22,215 genes down to a much smaller group of features that contributed to the classification task. With fewer noisy variables, the model became easier to train and behave more reliably. We then evaluated three SVM kernels, namely Linear, RBF, and Polynomial, each combined with two different MFO population sizes. Among all these trials, the Polynomial kernel produced the strongest performance on the test set, reaching an accuracy of 0.84 and an F1-score of 0.85. The Linear and RBF kernels also showed noticeable improvements once the features had been reduced, but the Polynomial configuration consistently stood out as the most stable across different settings. Overall, these findings reinforce the idea that careful feature selection plays an important role in gene-expression studies, and that pairing MFO with SVM can lead to more reliable prediction results. This study is limited to a binary diagnosis (diagnosed vs. undiagnosed) and does not yet distinguish between subtypes like SCLC or NSCLC. Future studies could try combining MFO with other optimization methods, like Particle Swarm or Grey Wolf Optimization, or to test different optimization algorithms to improve the feature selection results.

REFERENCES

[1] M. Abdul, R. Wahid, A. Nugroho, and A. H. Anshor, "Prediksi Penyakit Kanker Paru-Paru Dengan Algoritma Regresi Linier," *Bulletin of Information Technology (BIT)*, vol. 4, no. 1, pp. 63–74, 2023, doi: 10.47065/bit.v3i1.

[2] World Health Organization, "Lung cancer," Fact sheet, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/lung-cancer

[3] O. D. Asmara et al., "Lung Cancer in Indonesia," *Journal of Thoracic Oncology*, vol. 18, no. 9, pp. 1134–1145, 2023, doi: 10.1016/j.jtho.2023.06.010.

[4] A. N. Khudori and M. S. Haris, "Implementasi Decision Tree Untuk Prediksi Kanker Paru-Paru," *Jurnal Riset Sistem Informasi dan Teknik Informatika (JURASIK)*, vol. 9, no. 1, pp. 94–106, 2024.

[5] N. Fhira and I. Kurniawan, "Classification of Non-Small Cell Lung Cancer Based on Gene Expression in Cases of Smokers and Non-Smokers Using Ensemble Methods with Statistical Based Feature Selection," *Journal of Computer Science*, pp. 913–927, 2022, doi: 10.3844/jcssp.2022.913.927.

[6] A. M. Anasin, N. Fhira, and I. Kurniawan, "Implementation of PSO-SVM on Gene Expression Data for Lung Cancer Identification in Smoker Person," in *Proc. Int. Conf. Smart Computing, IoT and Machine Learning (SIML)*, 2024, pp. 1–4, doi: 10.1109/SIML61815.2024.10578108.

[7] K. N. Azizah, F. Nhita, and I. Kurniawan, "Model Klasifikasi Berbasis Ekspresi Gen Non-Small Cell Lung Carcinoma (NSCLC) pada Wanita Bukan Perokok Menggunakan Metode Ensemble," *Jurnal Penelitian Informatika*, vol. 1, pp. 1–7, 2023, doi: 10.25124/logic.v1i1.6489.

[8] A. Otniel, N. Fhira, and I. Kurniawan, "Identification of Lung Cancer in Smoker Person Using Methods Based on Gene Expression Data," in *Proc. 5th Int. Conf. Computer and Informatics Engineering (IC2IE)*, 2022, pp. 1–5.

[9] A. G. Hussien, M. Amin, and M. A. El Aziz, "A comprehensive review of moth-flame optimisation: variants, hybrids, and applications," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 32, no. 4, pp. 705–725, 2020, doi: 10.1080/0952813X.2020.1737246.

[10] M. Rejeki et al., "Diagnosis dan Prognosis Kanker Paru, Probabilitas Metastasis dan Upaya Prevensinya," in *Proc. The 12th University Research Colloqium*, 2020, pp. 1–6.

[11] B. S. C. Putra, I. Tahyudin, B. A. Kusuma, and K. N. Isnaini, "Efektivitas Algoritma Random Forest, XGBoost, dan Logistic Regression dalam Prediksi Penyakit Paru-paru," *Techno.COM*, vol. 23, no. 4, pp. 909–922, 2024.

[12] M. Talaat, A. S. Alsayyari, M. A. Farahat, and T. Said, "Moth-Flame Algorithm for Accurate Simulation of a Non-Uniform Electric Field in the Presence of Dielectric Barrier," *IEEE Access*, vol. 7, pp. 3836–3847, 2019, doi: 10.1109/ACCESS.2018.2889155.
[13] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015, doi: 10.1016/j.knosys.2015.07.006.
[14] P. Fremmuzar and A. Baita, "Uji Kernel SVM dalam Analisis Sentimen Terhadap Layanan Telkomsel di Media Sosial Twitter," *Komputika: Jurnal Sistem Komputer*, vol. 12, no. 2, pp. 1–10, 2023, doi: 10.34010/komputika.v12i2.9460.
[15] S. Rahmaeda and R. Prathivi, "Komparasi Metode SVM dan Logistic Regression untuk Klasifikasi Hipotesa Penyakit Kanker Paru Paru Berdasarkan Gejala Awal," *KESATRIA: Jurnal Penerapan Sistem Informasi*, vol. 6, no. 1, pp. 1–9, 2025.
[16] A. Afinda, A. M. Karimah, A. Kurniawan, and I. Kurniawan, "Gated Recurrent Unit with SMILES2Vec-based Descriptor for Predicting Drug Side Effects: Case Study of Hepatobiliary Disorders," in *Proc. Int. Conf. Data Sci. Appl. (ICoDSA)*, 2023, pp. 426–431, doi: 10.1109/ICoDSA58501.2023.10276594.
[17] M. Abdulla and M. T. Khasawneh, "G-Forest: An ensemble method for cost-sensitive feature selection in gene expression microarrays," *Artificial Intelligence in Medicine*, vol. 108, 2020, doi: 10.1016/j.artmed.2020.101941.
[18] A. M. Alharthi, M. H. Lee, and Z. Y. Algamal, "Gene selection and classification of microarray gene expression data based on a new adaptive L1-norm elastic net penalty," *Informatics in Medicine Unlocked*, vol. 24, 2021, doi: 10.1016/j.imu.2021.100622.
[19] A. Ali, Z. Khan, and S. Aldahmani, "Optimized feature selection in high-dimensional gene expression data using weighted differential gene expression analysis," *Applied Soft Computing*, vol. 180, 2025, doi: 10.1016/j.asoc.2025.113329.
[20] H. M. Alshamlan, G. H. Badr, and Y. A. Alohali, "Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification," *Computational Biology and Chemistry*, vol. 56, pp. 49–60, 2015, doi: 10.1016/j.compbiolchem.2015.03.001.
[21] M. Alzaqebah et al., "Memory based cuckoo search algorithm for feature selection of gene expression dataset," *Informatics in Medicine Unlocked*, vol. 24, 2021, doi: 10.1016/j.imu.2021.100572.
[22] R. Aziz, C. K. Verma, and N. Srivastava, "A novel approach for dimension reduction of microarray," *Computational Biology and Chemistry*, vol. 71, pp. 161–169, 2017, doi: 10.1016/j.compbiolchem.2017.10.009.
[23] I. A. Baba, M. B. Mohammed, K. B. Jillahi, A. Umar, and H. T. Hendi, "Robust correlation feature selection based support vector machine approach for high dimensional datasets," *Results in Control and Optimization*, vol. 21, p. 100609, 2025, doi: 10.1016/j.rico.2025.100609.
[24] M. C. Barbieri, B. I. Grisci, and M. Dorn, "Analysis and comparison of feature selection methods towards performance and stability," *Expert Systems with Applications*, vol. 249, 2024, doi: 10.1016/j.eswa.2024.123667.
[25] T. Cai et al., "Multi-Label Feature Selection Based on Improved Ant Colony Optimization Algorithm with Dynamic Redundancy and Label Dependence," *Computers, Materials and Continua*, vol. 81, no. 1, pp. 1157–1175, 2024, doi: 10.32604/CMC.2024.055080.
[26] Z. Chen et al., "Feature selection may improve deep neural networks for the bioinformatics problems," *Bioinformatics*, vol. 36, no. 5, pp. 1542–1552, 2020, doi: 10.1093/bioinformatics/btz763.
[27] S. Chormunge and S. Jena, "Correlation based feature selection with clustering for high dimensional data," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 542–549, 2018, doi: 10.1016/j.jesit.2017.06.004.
[28] L. Y. Chuang, C. H. Yang, K. C. Wu, and C. H. Yang, "A hybrid feature selection method for DNA microarray data," *Computers in Biology and Medicine*, vol. 41, no. 4, pp. 228–237, 2011, doi: 10.1016/j.compbiomed.2011.02.004.
[29] A. Dabba, A. Tari, S. Meftali, and R. Mokhtari, "Gene selection and classification of microarray data method based on mutual information and moth flame algorithm," *Expert Systems with Applications*, vol. 166, 2021, doi: 10.1016/j.eswa.2020.114012.
[30] M. Dashtban and M. Balafar, "Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts," *Genomics*, vol. 109, no. 2, pp. 91–107, 2017, doi: 10.1016/j.ygeno.2017.01.004.
[31] C. Li, J. Zhou, K. Du, and D. Dias, "Stability prediction of hard rock pillar using support vector machine optimized by three metaheuristic algorithms," *International Journal of Mining Science and Technology*, vol. 33, no. 8, pp. 1019–1036, 2023, doi: 10.1016/j.ijmst.2023.06.001.
[32] T. I. A. Mohamed and A. E. S. Ezugwu, "Enhancing Lung Cancer Classification and Prediction With Deep Learning and Multi-Omics Data," *IEEE Access*, vol. 12, pp. 59880–59892, 2024, doi: 10.1109/ACCESS.2024.3394030.
[33] S. Osama, H. Shaban, and A. A. Ali, "Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review," *Expert Systems with Applications*, vol. 213, p. 118946, 2023, doi: 10.1016/j.eswa.2022.118946.
[34] A. P., S. K. TV, M. Khan, M. M. Su'ud, M. M. Alam, and S. Mallik, "Ant Colony Optimization for feature selection in breast cancer classification," *Egyptian Informatics Journal*, vol. 32, p. 100847, 2025, doi: 10.1016/j.eij.2025.100847.
[35] A. Yaqoob et al., "A Review on Nature-Inspired Algorithms for Cancer Disease Prediction and Classification," *Mathematics*, vol. 11, no. 5, 2023, doi: 10.3390/math11051081.
[36] J. Zhou, S. Huang, and Y. Qiu, "Optimization of random forest through the use of MVO, GWO and MFO in evaluating the stability of underground entry-type excavations," *Tunnelling and Underground Space Technology*, vol. 124, p. 104494, 2022, doi: 10.1016/j.tust.2022.104494.