

Parametric Faults Detection in Analog Circuits using Variable Ranking-based Feature Selection Method and Optimized SVM Model

G. Puvaneswari 

Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Civil Aerodrome Post, Avinashi Road, Peelamedu, Coimbatore, 641014, Tamil Nadu, India, puvaneswari@cit.edu.in

Abstract: This work proposes an optimized support vector model and a variable ranking-based test node selection approach for identifying parametric faults in analog circuits using a fault dictionary. Test node selection is essential for fault dictionary-based fault detection to reduce the dimensionality and test process complexity. To determine an appropriate set of test nodes, a feature selection technique based on variable ranking is used, as it is computationally efficient and involves sorting and score estimation. In the proposed method, test nodes are ranked using a score function based on data variability, where the nodes with the highest data variability are assigned the highest rank. This ranking ensures that the most informative test nodes are prioritized for fault detection. An optimized support vector model is used for fault diagnosis to improve classification accuracy. The results show the effectiveness of this approach. The performance of the proposed method is validated by measuring the fault detection accuracy on benchmark circuits.

Keywords: analog circuit, fault diagnosis, test nodes, support vector machine, machine learning, feature selection

1. INTRODUCTION

Fault diagnosis in analog circuits is defined as the process of identifying whether the circuit under test (CUT) is faulty or not, and in case of faulty it aims to locate the faulty component. Fault diagnosis in analog circuits is divided into the categories of simulation before testing (SBT) and simulation after testing (SAT). In simulation before testing, the CUT is simulated using simulation software before a practical or real-time test is carried out. The CUT response is recorded for different fault cases of the components of the CUT, variations of the manufacturing process parameters and for different input conditions. The SBT technique helps the designer to analyze and solve problems related to the design before production or manufacturing. In simulation after testing, the simulation is performed once the design is available in the form of a physical product. Simulation and testing is carried out to understand the characteristics of a physical system under different operating conditions. SAT also helps the designer to analyze and understand the problems in the physical system and optimize the system design for better performance. Faults in an analog circuit are classified into soft faults or parametric faults and hard faults or catastrophic faults. Soft or parametric faults are defined as parameter deviation outside the tolerance limit that causes degradation in system performance or inaccurate response.

Hard or catastrophic faults are open circuit and short circuit faults. These faults completely override the function of the CUT and are responsible for the CUT's incorrect response. Therefore, the identification of these faults in analog circuits is crucial as these faults lead to the destruction of the circuit components and eventually the circuit. Numerous techniques have been developed to diagnose soft and hard faults, and automation of fault diagnosis is necessary to shorten the time for diagnosis and improve fault detection accuracy. In recent years, machine learning algorithms have been used to identify and classify fault types in analog circuits. In [1], fault detection is performed based on a convolutional neural network. Fast Fourier transform and principal component analysis are used to create fault dictionaries with reduced dimensions. SBT methods are generally based on a fault dictionary. The fault dictionary of an analog circuit contains simulation results of a CUT with a faulty component and its value as well as the CUT's response in terms of voltage level, gain, bandwidth, current level, etc. In [2], a deep learning-based technique is used for fault diagnosis and prediction. It also explains the challenges of deep learning-based methods in terms of unbalanced data handling, data fusion and fault types. Support vector machine (SVM)-based fault diagnosis methods are proposed in [3]. SVMs are often used to classify analog circuit faults as they can handle larger amounts of data

and provide higher accuracy and faster predictions. In [4], a semi-supervised SVM method is used to classify soft faults in analog circuits and a manifold learning algorithm is used for feature extraction from the time domain response of the CUT and for dimensionality reduction.

The biggest challenge in analog circuit fault diagnosis is the test node selection for testing. This is usually done in SBT approaches to reduce dimensions and test time. Test node selection in [5] is done by considering factors such as measurement accuracy, tolerance of the components, and the diagnostic method used. Frequency domain characteristics are used as parameters for fault diagnosis. Frequency domain testing is performed in [6]. The proposed method uses fault models defined for active and passive components and utilizes basic circuit laws to define the faulty element parameter. Wavelet transforms and generalized discriminant analysis technique based fault diagnosis feature extraction is presented in [7]. A generalized multiple kernel SVM is used to identify fault classes. To optimize the parameters of SVM, the particle swarm optimization (PSO) method is used. In [8], a correlation algorithm combined with SVM is used to diagnose soft faults in nonlinear analog circuits. Canonical correlation algorithm (CCA) is used to improve the feature correlation and principal component analysis is used for dimension reduction. SVM is used for fault classification. In [9] and [10], the application of machine learning and artificial intelligence algorithms to the problem of fault diagnosis is discussed. They clearly explain the need and importance of machine learning algorithms in the development of end-to-end diagnostic procedures. Discrete wavelet transform and SVM-based fault diagnosis techniques in machine health monitoring are explained in [11]. Fault detection in control systems is discussed in [12]. It explains the use of SVM to classify faults and the use of principal component analysis for dimension reduction. Kernel-based SVM is discussed in [13]. Kernel functions such as the radial basis function, Naïve Bayes, and linear and polynomial functions are explained and applied to machinery fault detection. The diagnosis is performed in three steps: data acquisition, feature extraction, and fault classification. Continuous wavelet transform is used for feature extraction and kernel-based SVM is used for fault classification. In [14]-[22], the significance and use of SVM for various classification problems is discussed. The advantage of using SVM with kernel features is shown. It is found that SVM with kernels can handle non-linear decision boundaries and that overfitting in data processing can be avoided by controlling the regularization parameters of SVM. Overfitting occurs when a machine learning model cannot accurately predict new data. Optimal regularization parameters are determined by genetic algorithms [19], holoentropy life choice optimization [21], PSO optimization [20], and Bayesian algorithms [22]. It can be observed that by tuning the SVM parameters, the accuracy of the SVM model can be improved to achieve optimal results.

The proposed analog circuit fault diagnosis uses an optimized SVM for fault classification and test node selection based on the statistical parameters of the circuit response to minimize the test time. Section 1 of this paper introduces the concept of fault diagnosis in analog circuits and the existing

methods. Section 2 explains the proposed fault diagnosis method and section 3 presents the results and conclusions. Section 4 concludes the presented work.

2. ANALOG CIRCUIT FAULT DIAGNOSIS PROCEDURE

Fault diagnosis in analog circuits is still an open problem and more new methods need to be developed to achieve the goals of diagnosis. The main objectives of analog circuit fault diagnosis are to identify the faulty condition of a CUT, locate the faulty component, and determine the extent of the deviation in the component that causes an anomaly in the CUT's response. Fig. 1 shows the stages of fault detection and localization in analog circuits.

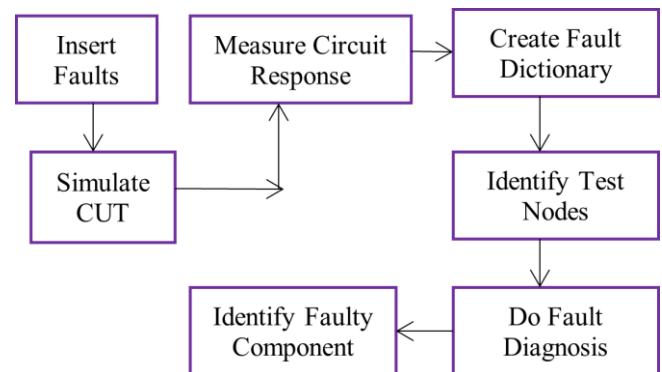


Fig. 1. Analog circuit fault diagnosis procedure.

An analog circuit is usually tested using a technique called fault dictionary-based fault diagnosis. In this technique, a dictionary of fault signatures is created for various known circuit fault conditions. Circuit response to different fault conditions is measured to obtain the fault signatures. The behavior of the circuit is measured by injecting known faults such as open circuit, short circuit, and component degradation, into the CUT. The fault dictionary is created by storing the measured or simulated responses of the circuit for each fault condition. The entry corresponding to a specific fault condition is called a fault signature in dictionary-based techniques. When a fault is suspected, the CUT's response is measured and compared with the fault signatures stored in the dictionary. To identify the closest match, the similarity between the measured response and each fault signature in the dictionary is calculated using appropriate measures. Once the fault has been detected, the faulty component is identified using additional diagnostic methods. Although fault dictionary-based techniques have proven to be powerful, they face many challenges, such as a high-dimensional fault space due to a wider range of fault types, measurement variability due to factors such as component tolerances, and instrumentation inaccuracies, discrepancies between simulated fault signatures and actual measurements, fault coverage, complexity of the CUT, and time required to create and update fault dictionaries to account for circuit design changes and fault behavior. Therefore, to improve the effectiveness of fault dictionary-based fault diagnosis in analog circuits, diagnostic techniques such as model-based analysis and machine learning-based fault detection are preferred.

A. Test node selection

The test node selection refers to the process of identifying suitable nodes for fault diagnosis and testing CUT functionality. The test node selection must take into account accessibility, maximum fault coverage, cost, test equipment requirements, test time, and valuable diagnostic information. Testing analog circuits is becoming more and more difficult due to the increasing complexity of the circuits. Test costs play an important role in determining overall manufacturing costs and product costs. Test time is one of the factors that determine test costs. Test time is proportional to the degree of complexity of the CUT and the number of nodes used for testing. The amount of data required for testing is also determined by the degree of complexity of a CUT and the number of test nodes. This shows the need for test node selection in fault diagnosis. Test node selection is therefore about identifying suitable nodes in order to carry out fault diagnosis at the lowest possible cost. Fault coverage is the decisive component in the selection of test nodes. It is defined as the ratio between the number of successfully detected faults and the total number of faults.

The test node selection in this proposed work is performed using the variable ranking method. Variable ranking is defined as the process of ranking the test nodes or variables based on the feature relevance measure (scoring function). In the variable ranking method, the highest ranked (rank 1) variables are determined on the basis of the feature relevance measure or the variable relevance measure. It is used to identify and remove test nodes that are least important from the set of nodes available for the test. The variables with the highest rank (rank 1) are selected for fault classification. This method is preferred here over the other complicated methods as it involves simple computation and it is a straightforward process of ordering or ranking the test nodes according to the measure. In the proposed work, the variables are ranked based on the standard deviation measure. The standard deviation is used to measure the dispersion of node voltages in the fault dictionary dataset. A higher value of standard deviation indicates a high variability of the data and is therefore preferred when localizing faults. The block diagram shown in Fig. 2 illustrates the procedure for test node selection.

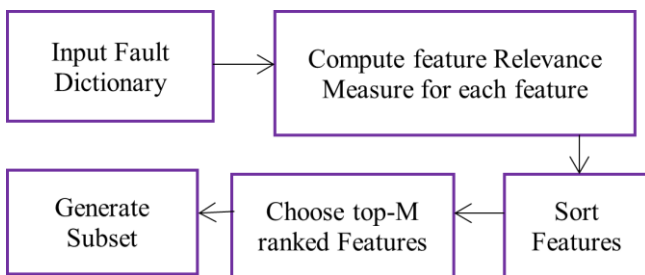


Fig. 2. Test node selection.

Fig. 2 explains the process of test node selection using the variable ranking method. For the given fault dictionary of a CUT, the standard deviation for each test node (feature) is calculated to measure the data variability, and the features are ranked according to the standard deviation value. This means that features with a lower standard deviation are ranked lower. In this method of feature or test node selection, the

rank is calculated by looking at individual nodes. It neglects the combination of nodes, i.e. the measurement of data variability considering two or more nodes together. This means that there could be test nodes that data variability when considered together, but a lower variability when considered individually. At the same time, the nodes could have higher data variability when considered individually, but a high correlation when considered together. To avoid this, the proposed method finds the subset of test nodes for fault diagnosis by considering combinations of features to evaluate data variability and for improved performance. By considering subsets of features, the combination of feature ranking can potentially capture interactions between features, which improve the predictive performance and provides a more thorough assessment of feature relevance. However, due to the combinatorial explosion of possible feature combinations, it is slower than single feature ranking, especially for large feature sets. Compared to single feature ranking, it is more complicated to implement and interpret, requires more computational resources and may not be practical for datasets with many features. To summarize, ranking a single feature is usually faster and computationally more efficient than ranking a combination of features. On the other hand, a combination of features can provide better predictive performance by capturing the interactions between the features.

Let N be the input fault dictionary with n fault cases and m test nodes or features (node voltages) and represented as $N = [N1, N2, N3, N4, \dots Nm]$, where Ni is the i^{th} feature vector of length n . The standard deviation σ_i of the i^{th} feature is calculated as follows:

$$\sigma_i = \sqrt{\frac{1}{n} \sum_{j=1}^n (N_{ij} - \bar{N}_i)^2} \quad (1)$$

where N_{ij} is the j^{th} sample of the i^{th} feature and \bar{N}_i is the mean value of the i^{th} feature. Let M be the target variable. The scoring function is defined as:

$$\text{score}(i) = \sigma(i, M), \text{ for } i = 1, 2, 3, \dots m \quad (2)$$

Test nodes with a higher standard deviation have a higher score, indicating greater importance in the test node selection process. The features are ranked based on the score:

$$R(m) = \text{argsort}(\text{score}(1), \text{score}(2), \dots \text{score}(m)) \quad (3)$$

and the feature subset is obtained with x features:

$$Fs = R(m)[x] \quad (4)$$

For m number of features, the computation time for single feature ranking is approximately:

$$O(mT_s) \quad (5)$$

where T_s is the time required to determine a single feature's relevance and for ranking a combination of features it is approximately:

$$O(2^m T_c) \quad (6)$$

where T_c is the time required to evaluate a single subset of features. In an exhaustive search, both the number of the subsets and the computation time increase exponentially with the number of features.

B. Fault classification

The faults in the proposed work are identified and classified using a SVM model. Machine learning-based classification algorithms use training data to predict the class of new data points. SVM is a widely used and preferred supervised learning model in machine learning-based classification problems. SVM classifies faults by constructing an optimal hyper plane that maximizes the margin between faulty and non-faulty classes. For the given training dataset:

$$\{(x_i, y_i)\}_{i=1}^N,$$

where $x_i \in R^n$ represents the test node measurements and $y_i \in \{-1, 1\}$ indicates the class labels, the decision boundary is defined by:

$$f(x) = w^T x + b \quad (7)$$

where w is a vector normal to the hyper plane and b is the offset. The SVM optimization problem [23] is formulated as:

$$\min_{w,b} \frac{1}{2} ||w||^2 \quad (8)$$

subject to the constraints:

$$y_i(w^T x_i + b) \geq 1, \forall i \quad (9)$$

For nonlinear separable data, the soft margin SVM is formulated as:

$$\min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^N \xi_i \quad (10)$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (11)$$

where ξ_i is a slack variable and C is the regularization parameter that controls the trade-off between margin maximization and the classification error. In the case of nonlinear data, the kernel functions transform to a higher dimension space. The widely used nonlinear kernel functions are given by:

$$\text{radial basis function (rbf)} \quad k(x, y) = e^{-\gamma ||x-y||^2} \quad (12)$$

$$\text{sigmoid} \quad k(x, y) = \tanh(\gamma(x^T y) + r) \quad (13)$$

$$\text{polynomial} \quad k(x, y) = \gamma(x^T y + r)^d \quad (14)$$

where r is an additive bias term that controls the behavior of the kernel function and accordingly the decision boundary in SVM, γ is a hyper parameter that determines the range of influence of each training point, and d is the degree of the polynomial kernel that determines the flexibility of the decision boundary. In fault classification, the choice of kernel and the hyper parameter tuning have a significant impact on accuracy. The most important steps in the optimization and

evaluation of an SVM model are shown in the block diagram in Fig. 3. These include hyper parameter tuning, cross-validation, training the final model, and evaluating its performance on unseen data. The fault dictionary obtained from the test node selection process is the dataset used to train the SVM model. The hyper parameters of the SVM model such as kernel parameters, kernel types and regularization parameters are also optimized to improve the performance of the model. Cross-validation is used to evaluate the optimized hyper parameters and determine how well the model performs on unknown data. The test dataset is used to evaluate the performance of the trained SVM model and measure its generalization performance.

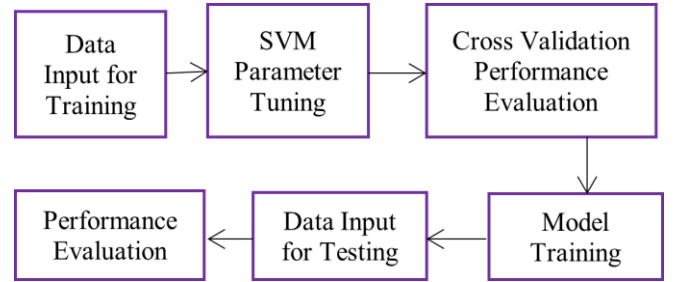


Fig. 3. SVM optimization.

PSO and Bayesian algorithms are generally used to optimize these hyper parameters. These algorithms reduce the computation time for an optimization problem, but only provide near-optimal solutions. Exhaustive search is another optimization method that searches through a specified range of hyper parameters to find the optimal parameters for better performance. The computation time of this method is higher than PSO and Bayesian algorithms because it searches through all possible solutions. But the exhaustive search always finds the best solution for an optimization problem [24]. Therefore, the proposed fault classification problem uses exhaustive search to optimize the SVM hyper parameters. Fig. 4 shows the flowchart of the exhaustive search method. The exhaustive search is a problem-solving technique in which the optimal solution is sought from all possible solutions that arise for all possible candidates. Although the search method is easy to implement, its computation time depends on the number of possible candidates. The computation time of the exhaustive search is given by $O(m)$, where m is the number of possible candidates. To reduce the computation time, the search space or the set of possible solutions is reduced. The search starts with the initial value of the SVM parameters and checks whether the desired classification accuracy is within the parameter range. The parameter values are updated and the algorithm selects the parameters for which the expected accuracy is achieved.

When evaluating the performance of a machine learning model, cross-validation is a technique used to minimize over fitting and under fitting errors. In cross-validation, the entire dataset is divided into k folds and the machine learning model is trained and tested with different training and test folds. The final performance score of a machine learning model is the average of all results.

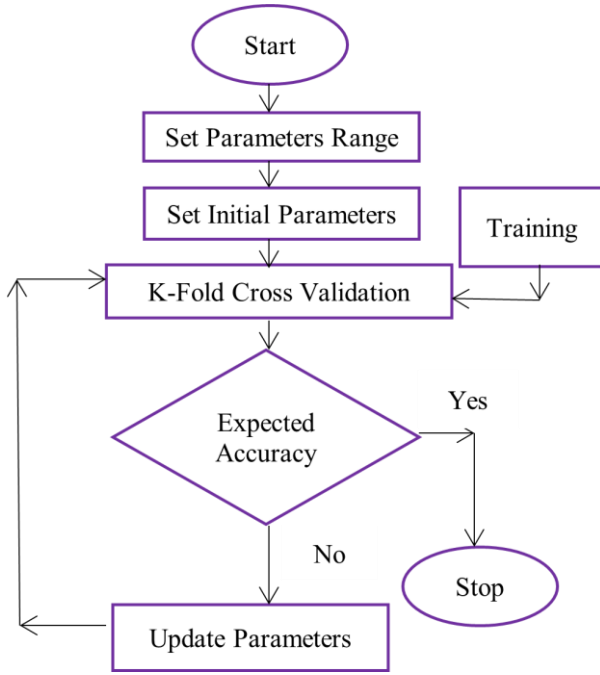


Fig. 4. Flowchart of exhaustive search.

3. RESULTS AND DISCUSSION

The performance of test node selection and fault identification and localization is verified using benchmark circuits such as the Sallen key BPF and the state variable filter (SVF). The results are presented for the different stages of fault diagnosis, taking into account parametric faults or soft faults. The fault dictionary for each component of CUT is created considering the component degradation above and below the tolerance limits. The acceptable level of component degradation in an analog circuit depends on the performance requirements. When there are strict requirements for accuracy and stability, even a small amount of degradation is not acceptable. In other situations, however, a certain amount of degradation may be acceptable as long as it does not negatively affect the overall performance. In this work, thirty percent to ninety percent degradation above and below the tolerance limits of the components are considered. The simulation is carried out and node voltages are recorded to create the fault dictionary. Test node selection is performed taking into account a single feature as well as a combination of features. In the case of a combination of features, fifty percent of the total test nodes are considered. This means that if the CUT contains ten nodes, the subset of test nodes selected for fault diagnosis contains five nodes and 2^5 combinations are to be evaluated. The decision to select fifty percent of the total nodes is based on an optimal balance between fault detection accuracy and computational efficiency, as there is always a trade-off between the number of test nodes and fault detection accuracy. In the test node selection process, test nodes are selected based on high fault observability to ensure that faults occurring at different locations in the circuit can be effectively distinguished. In addition, the nodes that contribute significantly to circuit performance variations are considered to ensure that the most sensitive nodes are selected while reducing the computation burden. The faults are detected using a hyper-tuned or

optimized SVM model. The performance of the optimized SVM model is evaluated in terms of accuracy. It is defined as the percentage of fault cases correctly classified by all instances.

A. Sallen key band pass filter

The Sallen key band pass filter (BPF) is a frequency selective network used to extract a specific range of frequencies from the input signal and is widely used in signal processing, communication and instrumentation systems. It consists of a total of 5 nodes, including the input and output nodes. The bandwidth of the filter can be adjusted using appropriate component values. It contains an operational amplifier, five resistors and two capacitors. Fig. 5 shows the circuit diagram of a Sallen key BPF with its nominal component values. The tolerance limit of all the resistances is assumed to be $\pm 5\%$ and for the capacitances it is assumed to be $\pm 1\%$. A 500 mV, 60 Hz sine wave is applied to the input node (node 1) and the node voltages (magnitudes) are measured. The test frequency is selected based on the frequency response characteristics of the filter. The fault dictionary is created based on these values measured for different fault conditions. The 60 Hz frequency was chosen as it is within the bandwidth of the filter (Fig. 6).

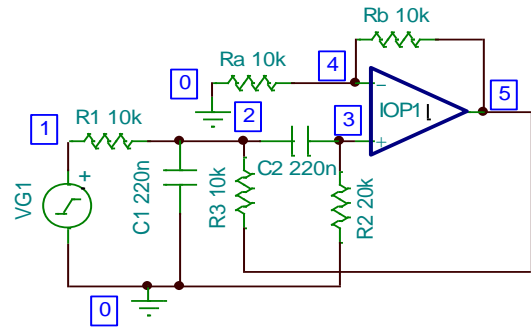


Fig. 5. Sallen key BPF.

The closed loop gain of the filter is given by:

$$A_{CL} = 1 + \frac{R_b}{R_a} \quad (15)$$

The frequency response characteristic is shown in Fig. 6.

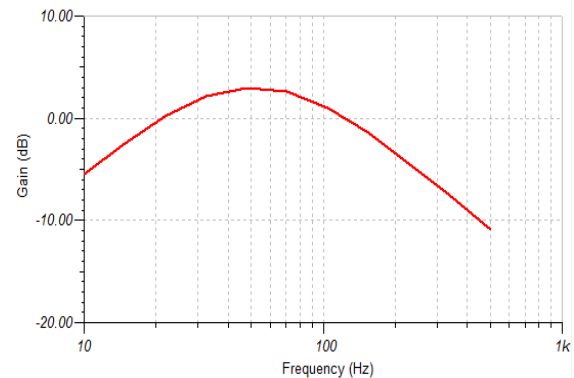


Fig. 6. Frequency response characteristics of BPF.

The center frequency at which the maximum response is achieved is given by:

$$f_o = \frac{1}{2\pi} \sqrt{\frac{R_3 + R_1}{C_1 C_2 R_1 R_2 R_3}} \quad (16)$$

The test node selection is based on the individual variable ranking, and the combinations of variable ranking and the variables with the highest data variability are used for fault diagnosis. The total number of nodes in the Sallen key BPF is 5. Without the input node 1, the total number of test nodes is 4. Fig. 7 shows the results for the test node selection with the test nodes of the BPF and the rank obtained for individual test nodes. The results are obtained using a single node or a feature ranking. It can be seen that node 5 (with rank 1) has the highest data variability. Therefore, it can be used for fault detection. The faults are identified and localized using an optimized SVM.

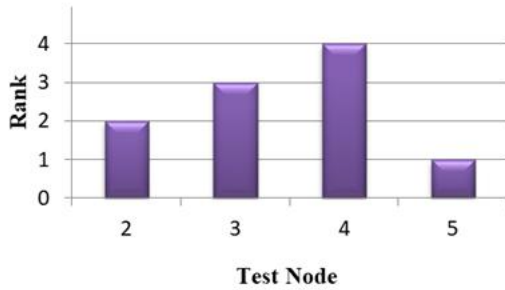


Fig. 7. BPF individual test node ranking.

The accuracy of fault classification obtained with an optimized SVM is shown in Fig. 8. The results are shown for all kernel functions. In the case of a non-optimized SVM, the regularization parameter is set to 1 and gamma is estimated from the number of test nodes used for testing and the variance of the test nodes. It can be observed that the optimized SVM shows better performance. Table 1 shows the results obtained for hyper parameter tuning using random search, Bayesian search with 100 iterations, and PSO with 100 iterations and a particle size of 10.

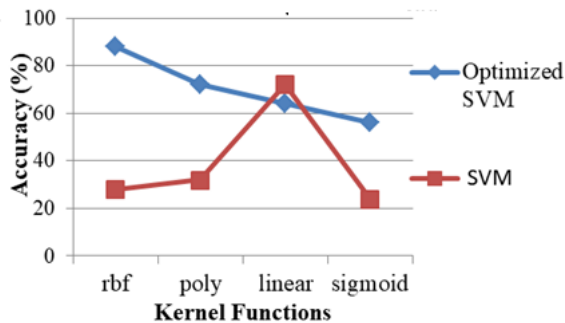


Fig. 8. BPF accuracy of classification single test node.

Table 1 shows that the exhaustive search achieves a 90 % classification accuracy at $C = 100$ and $\gamma = 0.0127$, since the exhaustive search tests all possible combinations of the hyper parameters of the SVM. In the random search, the hyper

parameters are selected at random, which can lead to suboptimal results due to the lack of systematic exploration of the search space. Bayesian search uses probability models to find the best hyper parameters, resulting in values that are close to the optimum.

Table 1. SVM hyper parameters for BPF - single node testing.

Optimization method	Fault detection accuracy		Hyper parameters
	Kernel	Accuracy [%]	
Random search	linear	65	$C = 100$
	rbf	80	$C = 100, \gamma = 0.012$
	poly	55	$C = 100, d = 2$
	sigmoid	80	$C = 100, \gamma = 0.012$
Bayesian search (number of iterations = 100)	linear	70	$C = 223.64$
	rbf	80	$C = 63.65, \gamma = 1.82$
	poly	75	$C = 6.01, d = 3$
	sigmoid	75	$C = 338.98, \gamma = 0.575$
PSO (number of iterations = 100, particle size = 10)	linear	65	$C = 412$
	rbf	80	$C = 921.4, \gamma = 6.9$
	poly	55	$C = 844.5, d = 2$
	sigmoid	70	$C = 307.3, \gamma = 0.12$
Exhaustive search	linear	65	$C = 100$
	rbf	90	$C = 100, \gamma = 0.0127$
	poly	70	$C = 1000, d = 2$
	sigmoid	55	$C = 1000, \gamma = 0.1$

PSO uses a population-based search inspired by swarm behaviour that efficiently explores the hyper parameter space to find near-optimal solutions. However, it does not guarantee the absolute best solution as it may converge to a local optimum. Fig. 9 shows the results for the test node selection considering fifty percent of the total test nodes for fault localization. Fig. 8 shows that test nodes 2 and 5 are suitable as test nodes as they have the highest data variability among the other test node combinations. It can also be observed that for single nodes, ranking is calculated for only four nodes, but when combinations of test nodes are considered, ranking has to be applied for six combinations. This increases the time required for ranking or selecting test nodes according to (5) and (6).

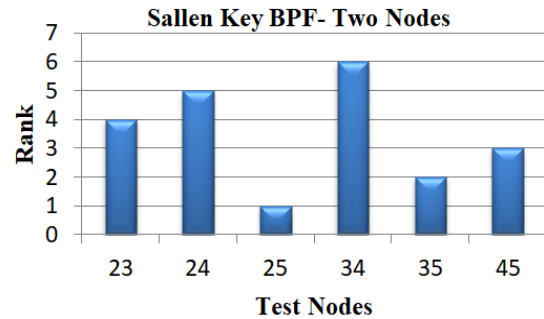


Fig. 9. BPF test node ranking.

Fig. 10 shows the fault detection accuracy achieved for the test of two nodes. It shows that with an optimized SVM and with rbf and poly kernels, a classification accuracy of 100 %

can be achieved when two nodes are used for testing. It can also be observed that the use of multiple nodes leads to better fault coverage.

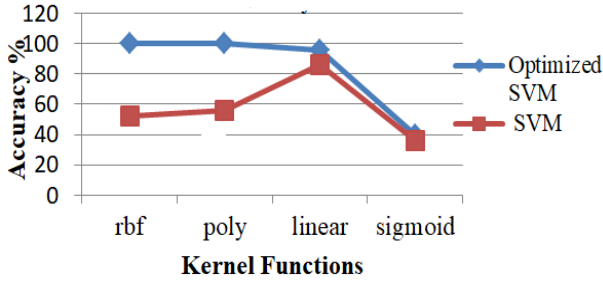


Fig. 10. BPF accuracy of classification - two test nodes.

Table 2. SVM hyper parameters for BPF.

Optimization method	Fault detection accuracy		Hyper parameters
	Kernel	Accuracy [%]	
Random search	linear	95	$C = 100$
	rbf	95	$C = 100, \gamma = 0.0085$
	poly	65	$C = 100, d = 3$
	sigmoid	80	$C = 100, \gamma = 0.0085$
Bayesian search (number of iterations = 100)	linear	92	$C = 1000$
	rbf	95	$C = 1.913, \gamma = 0.9$
	poly	90	$C = 86.32, d = 3$
	sigmoid	90	$C = 1000, \gamma = 0.0417$
PSO (number of iterations = 100, particle size = 10)	linear	96	$C = 831$
	rbf	95	$C = 200.42, \gamma = 6.2$
	poly	95	$C = 256, d = 3$
	sigmoid	85	$C = 218.57, \gamma = 0.111$
Exhaustive search	linear	100	$C = 1000$
	rbf	100	$C = 1000, \gamma = 0.00857$
	poly	95	$C = 1000, d = 2$
	sigmoid	45	$C = 1000, \gamma = 0.1$

Table 2 shows the results for classification accuracy and SVM hyper parameters values for different optimization methods. It shows that the fault detection accuracy improves as the number of test nodes increases, so that more fault cases can be identified. However, this also leads to a longer test

time. The random search selects the hyper parameters randomly, it may miss the optimal solution and due to the randomness, different runs may yield different results, making it less reproducible compared to other methods. However, the exhaustive search ensures that the same results are obtained every time with the same dataset and the same settings.

B. State variable filter

SVF use two operational amplifier integrators and an adder to simultaneously generate second order low pass filter (LPF), BPF, and high pass filter (HPF) responses. They use two operational amplifier integrators and an operational adder to obtain the filter responses simultaneously. The SVF configuration is shown in Fig. 11. The circuit is tested by applying a 1 V, 1 kHz sine wave to node 8 and the node voltages at different nodes of the SVF are measured under different faulty conditions of the components. A fault dictionary is created based on these measurements. The circuit is shown with nominal values of all the resistive components. It is assumed that all resistance components have a tolerance of $\pm 5\%$ and all capacitances have a tolerance of $\pm 1\%$. The high pass, band pass and LPF responses are obtained at nodes 2, 4, and 6, respectively, as shown in Fig. 11. The Laplace transform of the BPF output (V_{BP}) is given by:

$$V_{BP} = -\frac{1}{sR_3C_1} V_{HP} \quad (17)$$

The LPF response is represented by:

$$V_{LP} = -\frac{1}{sR_4C_3} V_{BP} \quad (18)$$

and the HPF response is given by:

$$V_{HP} = -V_i - V_{LP} + \alpha V_{BP} \quad (19)$$

where the attenuation coefficient:

$$\alpha = \left(1 + \frac{R_2}{R_1 \parallel R_5}\right) \left(\frac{R_6}{R_6 + R_7}\right) \text{ and } V_i = V_{G1} \quad (20)$$

The frequency response of the LPF is shown in Fig. 12.

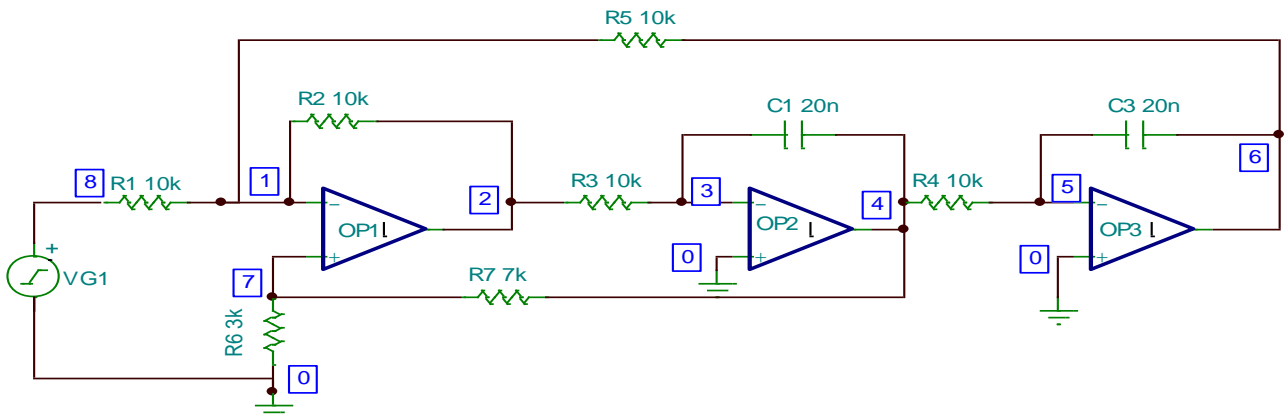


Fig. 11. State variable filter.

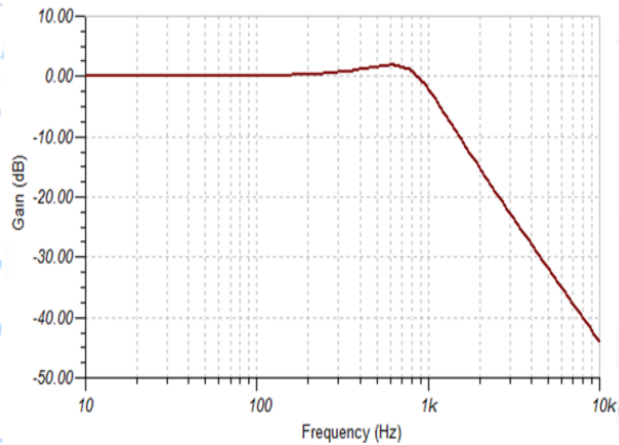


Fig. 12. Response of the LPF of the SVF.

The frequency response of the HPF of the SVF from Fig. 10 is shown in Fig. 13 and for the BPF of the SVF in Fig. 14.

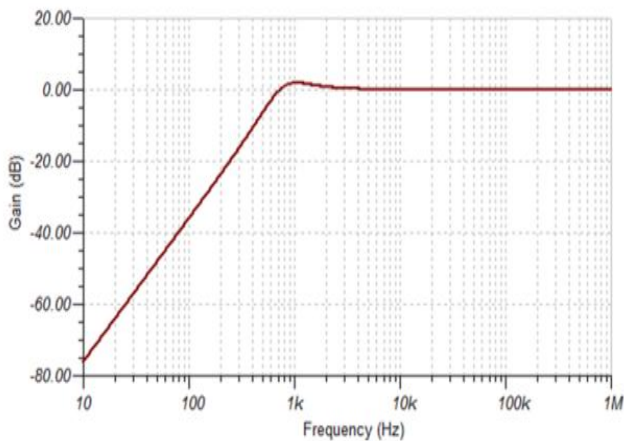


Fig. 13. Response of the HPF.

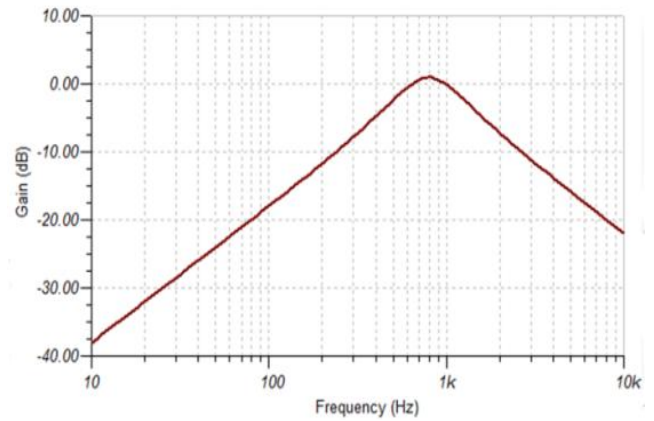


Fig. 14. Response of the BPF.

The result for the variable ranking when considering a single node is shown in Fig. 15. The node that has higher data variability is assigned rank 1. Node 6 has the highest rank (rank 1), i.e. it is the node with the highest data variability. However, more than one node is required to improve fault coverage.

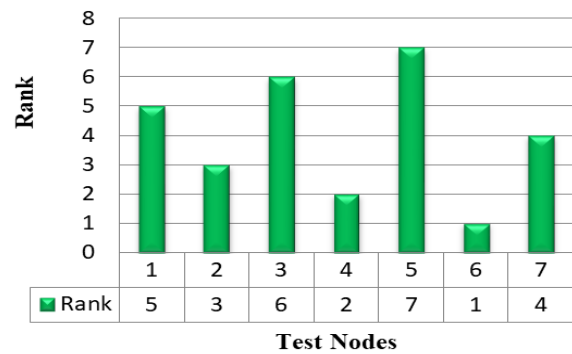


Fig. 15. SVF single test node ranking.

Fig. 16 shows the results of the variable ranking considering fifty percent of the total test nodes. The result shows that the highest ranking (rank 1) is obtained for test node 126.

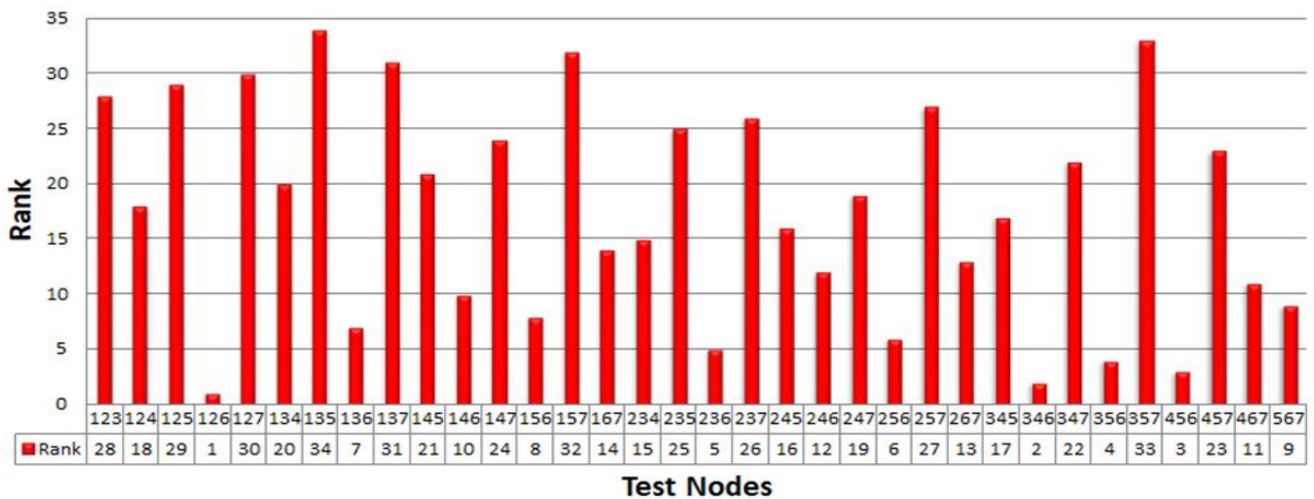


Fig. 16. SVF three test nodes ranking.

Therefore, the combination of test nodes 1, 2, and 6 is used for testing. Fig. 17 illustrates the fault detection accuracy for three test nodes, where a maximum accuracy of 90 % is achieved with the rbf kernel with $C = 1000$ and $\gamma = 0.046$.

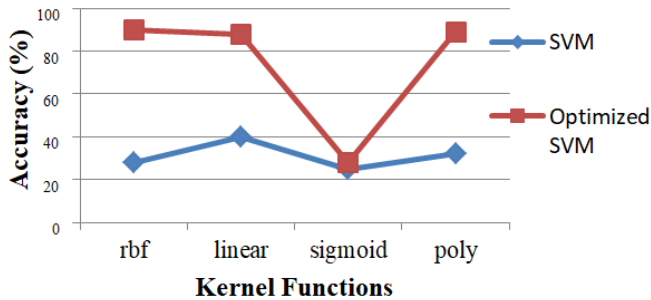


Fig. 17. SVF SVM classification accuracy.

Table 3 and Table 4 show the fault detection accuracy and the corresponding SVM hyper parameters for single and multiple test node scenarios. For single-node testing, the highest fault detection accuracy achieved by exhaustive search is 50 %. For multiple-node testing, the accuracy increases to a maximum of 90 %. This means that the classification accuracy is influenced by both the number of test nodes used for the tests and the SVM hyper parameters.

Table 3. SVF SVM hyper parameters - single node testing

Optimization method	Fault detection accuracy		Hyper parameters
	Kernel	Accuracy [%]	
Random search	linear	22.22	$C = 4.4$
	rbf	11.11	$C = 100, \gamma = 0.01$
	poly	11.11	$C = 35.93, d = 2$
	sigmoid	22.22	$C = 100, \gamma = 0.01$
Bayesian search (number of iterations = 100)	linear	33.33	$C = 1.179$
	rbf	16.67	$C = 3.08, \gamma = 0.243$
	poly	5.56	$C = 74.63, d = 2$
	sigmoid	16.67	$C = 1000, \gamma = 0.0013$
PSO (number of iterations = 100, particle size = 10)	linear	22.22	$C = 887.42$
	rbf	22.22	$C = 267, \gamma = 0.101$
	poly	16.67	$C = 256, d = 2$
	sigmoid	16.67	$C = 154.4, \gamma = 0.022$
Exhaustive search	linear	22.22	$C = 100$
	rbf	50	$C = 100, \gamma = 0.092$
	poly	50	$C = 1000, d = 2$
	sigmoid	22.22	$C = 1000, \gamma = 0.01$

The results show that the exhaustive search thoroughly evaluates all possible combinations of hyper parameters and that the search is systematic. For the same dataset and the same settings, it always yields the same results and is therefore absolutely deterministic. In the case of Bayesian optimization, search is based on probabilistic models and due to the randomness in the selection of the next best candidate, the reproducibility is somewhat lower. However, with proper settings, e.g. a higher number of iterations, reproducibility and better classification accuracy can be achieved.

Table 4. SVF SVM hyper parameters.

Optimization method	Fault detection accuracy		Hyper parameters
	Kernel	Accuracy [%]	
Random search	linear	22.22	$C = 12.9$
	rbf	50	$C = 100, \gamma = 0.034$
	poly	27.78	$C = 100, d = 2$
	sigmoid	27.78	$C = 100, \gamma = 0.01$
Bayesian search (number of iterations = 100)	linear	55.6	$C = 157.8$
	rbf	66.67	$C = 1000, \gamma = 0.207$
	poly	66.67	$C = 1000, d = 3$
	sigmoid	44.44	$C = 769.83, \gamma = 0.0086$
PSO (number of iterations = 100, particle size = 10)	linear	50	$C = 191.37$
	rbf	50	$C = 267.13, \gamma = 0.172$
	poly	37	$C = 152, d = 3$
	sigmoid	27.78	$C = 261.3, \gamma = 0.0158$
Exhaustive search	linear	88	$C = 100$
	rbf	90	$C = 1000, \gamma = 0.046$
	poly	89	$C = 1000, d = 3$
	sigmoid	28	$C = 1000, \gamma = 0.01$

4. CONCLUSION

This article proposes an optimized SVM-based classification technique along with a variable ranking mechanism for fault detection in analog circuits. Appropriate test nodes are selected from the collection of nodes in the CUT to reduce the time required for fault diagnosis. The data variability metric is used to select the test nodes, and the node subset with the highest data variability is given a higher rank. The node subset with the highest rank is used for fault detection. Test node selection takes into account both individual nodes and node combinations at the same time. The ranking method requires less time for single node rankings. However, a mix of nodes is selected for testing to achieve better performance. Certain test nodes may have low data variability when considered individually and high data variability when considered collectively. It has been found that optimized test node selection strategies can lead to better performance. Optimal support vector machines are used to classify faults. A comparison is made between SVM and the improved optimized SVM. It was found that a significant improvement is possible with the optimized SVM.

REFERENCES

- [1] Khalil, K., Eldash, O., Kumar, A., Bayoumi, M. (2020). Machine learning-based approach for hardware faults prediction. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67 (11), 3880-3892. <https://doi.org/10.1109/TCSI.2020.3010743>
- [2] Wang, H., Wei, J., Li, P. (2022). Research on fault diagnosis technology based on deep learning. *Journal of Physics: Conference Series*, 2187, 012041. <https://doi.org/10.1088/1742-6596/2187/1/012041>
- [3] Kim, M.-C., Lee, J.-H., Wang, D.-H., Lee, I.-S. (2023). Induction motor fault diagnosis using support vector machine, neural networks, and boosting methods. *Sensors*, 23 (5), 2585. <https://doi.org/10.3390/s23052585>

- [4] Wang, L., Tian, H., Zhang, H. (2021). Soft fault diagnosis of analog circuits based on semi-supervised support vector machine. *Analog Integrated Circuits and Signal Processing*, 108 (2), 305-315. <https://doi.org/10.1007/s10470-021-01851-w>
- [5] Hałgas, S. (2023). Soft fault diagnosis in linear circuits: Test selection and non-iterative identification procedure. *Measurement*, 217, 113061. <https://doi.org/10.1016/j.measurement.2023.113061>
- [6] Tadeusiewicz, M., Hałgas, S. (2022). A method for parametric and catastrophic fault diagnosis of analog linear circuits. *IEEE Access*, 10, 27002-27013. <https://doi.org/10.1109/ACCESS.2022.3157647>
- [7] Gao, T., Yang, J., Jiang, S. (2020). A novel incipient fault diagnosis method for analog circuits based on GMKL-SVM and wavelet fusion features. *IEEE Transactions on Instrumentation and Measurement*, 70, 3502315. <https://doi.org/10.1109/TIM.2020.3024337>
- [8] Li, Y., Zhang, R., Guo, Y., Huan, P., Zhang, M. (2020). Nonlinear soft fault diagnosis of analog circuits based on RCCA-SVM. *IEEE Access*, 8, 60951-60963. <https://doi.org/10.1109/ACCESS.2020.2982246>
- [9] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138, 106587. <https://doi.org/10.1016/j.ymssp.2019.106587>
- [10] Marino, R., Wisultschew, C., Otero, A., Lanza-Gutierrez, J. M., Portilla, J., de la Torre, E. (2020). A machine-learning-based distributed system for fault diagnosis with scalable detection quality in industrial IoT. *IEEE Internet of Things Journal*, 8 (6), 4339-4352. <https://doi.org/10.1109/JIOT.2020.3026211>
- [11] Goyal, D., Choudhary, A., Pabla, B. S., Dhami, S. S. (2020). Support vector machines based non-contact fault diagnosis system for bearings. *Journal of Intelligent Manufacturing*, 31, 1275-1289. <https://doi.org/10.1007/s10845-019-01511-x>
- [12] de Souza, D. L., Granzotto, M. H., de Almeida, G. M., Oliveira-Lopes, L. C. (2014). Fault detection and diagnosis using support vector machines - a SVC and SVR comparison. *Journal of Safety Engineering*, 3 (1), 18-29. <https://doi.org/10.5923/j.safety.20140301.03>
- [13] Al Tobi, M. A. S., Ramachandran, K. P., Al-Araimi, S., Pacturan, R., Rajakannu, A., Achuthan, C. (2022). Machinery faults diagnosis using support vector machine (SVM) and Naïve Bayes classifiers. *International Journal of Engineering Trends and Technology*, 70 (12), 26-34. <https://doi.org/10.14445/22315381/IJETT-V70I12P204>
- [14] Onel, M., Kieslich, C. A., Pistikopoulos, E. N. (2019). A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process. *AIChE Journal*, 65 (3), 992-1005. <https://doi.org/10.1002/aic.16497>
- [15] Gzar, D. A., Mahmood, A. M., Abbas, M. K. (2022). A comparative study of regression machine learning algorithms: Tradeoff between accuracy and computational complexity. *Mathematical Modelling of Engineering Problems*, 9 (5), 1217-1224. <https://doi.org/10.18280/mmep.090508>
- Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408, 189-215. <https://doi.org/10.1016/j.neucom.2019.10.118>
- [16] Zhang, S., Yin, Y., Liu, C., Li, J., Sun, X., Wu, J. (2023). Discrimination of wheat flour grade based on PSO-SVM of hyperspectral technique. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 302, 123050. <https://doi.org/10.1016/j.saa.2023.123050>
- [17] Gaye, B., Zhang, D., Wulamu, A. (2021). Improvement of support vector machine algorithm in big data background. *Mathematical Problems in Engineering*, 2021, 594899. <https://doi.org/10.1155/2021/5594899>
- [18] Manurung, J., Mawengkang, H., Zamzami, E. (2017). Optimizing support vector machine parameters with genetic algorithm for credit risk assessment. In *Journal of Physics: Conference Series*, 930, 012026. <https://doi.org/10.1088/1742-6596/930/1/012026>
- [19] Shaik Afzal, A. M. S. (2021). Optimized support vector machine model for visual sentiment analysis. In *2021 3rd International Conference on Signal Processing and Communication (ICSPC)*. IEEE, 171-175. <https://doi.org/10.1109/ICSPC51351.2021.9451669>
- [20] Josphineleela, R., Khan, M., Baskaran, R., Ul Haque, S. M. F., Shaik, A., Shankar S. S. (2023). Optimized support vector machine for big data classification NPC. *International Journal of Intelligent Systems and Applications in Engineering*, 11 (6S), 674. <https://ijisae.org/index.php/IJISAE/article/view/2904>
- [21] Elshewey, A. M., Shams, M. Y., El-Rashidy, N., Elhady, A. M., Shohieb, S. M., Tarek, Z. (2023). Bayesian optimization with support vector machine model for Parkinson disease classification. *Sensors*, 23 (4), 2085. <https://doi.org/10.3390/s23042085>
- [22] Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273-297. <https://doi.org/10.1007/BF00994018>
- [23] Nersisyan, S., Novosad, V., Galatenko, A., Sokolov, A., Bokov, G., Konovalov, A., Alekseev, D., Tonevitsky, A. (2022). ExhaustFS: Exhaustive search-based feature selection for classification and survival regression. *PeerJ*, 10, e13200. <https://doi.org/10.7717/peerj.13200>

Received February 24, 2024

Accepted March 18, 2025