CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 10 (2) (2025), Pages 698-710
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Health Insurance Claim Classification using Support Vector Machine with Velocity Pausing Particle Swarm Optimization

Luh Putu Dharma Jayanti, Syaiful Anam*, Safrizal Ardana Ardiyansa, and Natasha Clarissa Maharani

Mathematics Department, Faculty of Mathematics and Natural Science, Brawijaya University, Malang, Indonesia

Abstract

The classification of health insurance claims is a critical task for insurers in assessing potential risks and maintaining stability. However, predicting claim outcomes remains a challenging problem due to complex and high-dimensional data. This problem can be solved using Machine Learning (ML) to predict possible claim decisions. Support Vector Machine (SVM) is a ML model that has the ability to generalize the test data. Particle Swarm Optimization (PSO) can improve the SVM performance, but the solutions found by PSO are often trapped in local optima. Therefore, this article introduces a novel approach by applying Velocity Pausing PSO (VPPSO) strategy to optimize SVM in insurance claim classification, which has not been previously implemented in a previous study. Compared to SVM and SVM-PSO, the proposed SVM-VPPSO significantly reduces the risk of premature convergence and consistently produces better parameter results. SVM-VPPSO with linear kernel achieves a f_1-score of 90.17%, 90.16%, and 90.06% with 10, 20, and 30 particles, respectively. The linear kernel also performs better than RBF with a difference of 0.39% on the test data. The best configuration is SVM-Linear-VPPSO using 10 particles. This configuration achieves a computation time of 46.938 seconds, which is faster than that of SVM-Linear-VPPSO with 20 particles. The variance in computational time with 10 particles is 1.832s, better than 20 particles with a variance time of 37.909s. The proposed method offers a significant practical impact on real-world insurance, enabling early and accurate detection of health insurance claims, reducing fraudulent claims, and improving customer trust.

Keywords: Claim classification; health insurance; Support Vector Machine; Velocity Pausing Particle Swarm Optimization.

1 Introduction

Insurance is a financial agreement in which an individual or policyholder pays a premium to an insurance company in exchange for protection against specified risks [1]. Health insurance provides financial coverage for medical expenses such as surgeries, hospitalization, and treatment for policyholder injuries [2]. The party who pays the premium is called the policyholder, and the party responsible for providing this compensation is called the insurer [3]. By offering this, health insurance reduces the financial burden, allowing them to allocate resources to other essential needs [4]. Moreover, it also contributes to a greater sense of security, as the potential risks are absorbed by the insurance provider [5].

*Corresponding author. E-mail: syaiful@ub.ac.id

Submitted: February 04, 2025 Reviewed: May 23, 2025 Accepted: July 17, 2025
DOI: https://doi.org/10.18860/cauchy.v10i2.31914

According to Otoritas Jasa Keuangan, an insurance claim is a process through which the policyholder seeks compensation for the losses incurred by the insurer [5]. The risk factors that lead to health insurance claims depend on the insurance provider and the type of coverage. Both insurers and policyholders are exposed to risks related to claims, which affect competitiveness and lead to termination of insurance contracts [6]. Common risk factors in health insurance claims include the policyholder's age, sex, body mass index, number of dependents, lifestyle habits such as smoking, place of residence, and amount of premium paid. Claim classification plays a vital role in helping insurance companies determine whether a policyholder is likely to file a claim. Several factors can prevent policyholders from submitting claims, including incomplete health examination records [7], lack of transparency in information [8], and financial instability or liquidity problems within insurance companies [9].

The risks associated in claim classification drive insurance companies to predict potential claims through a classification process before making decisions. This approach is necessary to maintain the financial stability of the insurance company. Effective classification relies on identifying specific patterns within the data [10]. These patterns can be analyzed with various techniques, including an automated system designed to detect potential risk for claim classification [11]. This analysis enables companies to recognize similarities in policyholder data that may indicate a higher likelihood of filing a claim.

Manually predicting claim classifications is both challenging and time consuming. Machine Learning (ML) offers a promising solution by enabling automated insurance claims prediction, which can help estimate costs and reduce costs for organizations [12]. Various models can be applied for this task, including K-Nearest Neighbors (KNN), Naive Bayes (NB), Artificial Neural Networks (ANN), and Support Vector Machine (SVM). Among these models, SVM is a widely used pattern recognition model, achieving a 98% success rate in geotechnical engineering applications [13]. It has demonstrated excellent generalization performance in various domains [14]. For example, in healthcare to diagnose asthma, SVM achieves the highest accuracy of 98.59%, outperforming KNN, NB, and ANN [15]. In the context of fermentation, SVM also shows strong performance and generalization ability [16]. This capability allows SVM to classify new data that were not included in the training set [17].

SVM involves several key parameters, including the regularization parameter (C), the kernel type, and the kernel-specific parameter (γ). The regularization parameter is particularly important to maintain consistency of the model, with Gaussian RBF kernels known to perform well on noisy data [18]. However, determining the optimal values for these parameters is often done through trial and error, which can be time consuming and inefficient [19]. To address this, optimization algorithms are employed to automatically find the best parameter settings based on performance.

One of such optimization algorithms is Particle Swarm Optimization (PSO). PSO is a metaheuristic algorithm known for its flexibility, ease of implementation, convergence stability, and efficiency [20], [21]. When combined with SVM, PSO can effectively optimize parameter selection without compromising classification accuracy [22]. It also enables global optimization without relying on derivative information. A study by Anam et al. [23] applied this combination to the classification of health insurance claims and demonstrated that integrated SVM with PSO significantly outperforms standard SVM in performance. However, the previous study only explored the standard PSO as an optimization method and no modifications to the PSO (e.g., velocity pausing mechanisms) were explored. This opens a research gap for introducing enhanced variants like VPPSO to improve exploration and exploitation balance.

Although standard PSO can improve performance over standard SVM, it is known to suffer from premature convergence and become trapped in local optima, which affects the quality of parameter tuning [24]. This issue arises from a low-quality initial particle swarm and the algorithm's tendency to settle in local optima [25]. To address this, several modifications have been proposed in metaheuristic algorithms, including crossover [26], chaotic initialization [27], and the velocity pausing strategy [28].

Velocity Pausing Particle Swarm Optimization (VPPSO) is an improved version of PSO that implements the velocity pausing strategy. Previous studies state that VPPSO offers an improved balance between exploration and exploitation during the search process [29]. It employs time-varying inertia to achieve a more stable and balanced optimization, and introduces a third movement strategy that allows particles to retain their velocity from the previous iteration. Furthermore, VPPSO divides the population into two groups to preserve diversity [28]. Although VPPSO has shown promise, it has not yet been implemented in the insurance case. Considering the importance of accurate claim classification, this research proposes the integration of SVM-VPPSO as a novel approach. The rationale behind combining SVM and VPPSO lies in leveraging the high generalization capability of SVM with the enhanced search ability of VPPSO to achieve better parameter tuning and, consequently, improved classification accuracy. Based on this background, SVM-VPPSO is proposed for the classification of insurance claims. In general, the main contribution or novelty of this article is as follows.

1. This research introduces a novel implementation of VPPSO to optimize SVM for the classification of health insurance claims, filling a research gap where VPPSO has not been previously applied in this domain.
2. This research study systematically compares SVM configurations using linear and RBF kernels, with and without optimization, across various particle sizes to identify the most effective model configuration in insurance claim classification.

2 Methods

This section explains the dataset that will be used, the research steps, the explanation about the SVM and PSO algorithm, the proposed VPPSO, and the parameter configuration.

2.1 Research Dataset

The dataset used in this study is a Kaggle-sourced health insurance claim prediction. It comprises seven attributes, which are age, sex, Body Mass Index (BMI) of the policyholder, number of children, smoking status, region, and cost of the claim. The dataset encompasses seven features and 1,338 entries, with two categorical output classes, claim and non-claim. A sample of the insurance claim prediction dataset is shown in Table 1.

Table 1: Sample of Insurance Claim Prediction Dataset

No	Age	Sex	BMI	Children	Smoker	Region	Charges	Insurance Claim
1	19	0	27.90	0	1	3	16884.92	1
2	18	1	33.77	1	0	2	1725.52	1
3	28	1	33.00	3	0	2	4449.40	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1338	61	0	29.07	1	1	1	29141.36	1

The explanation features of the dataset is as follows. Age denotes the policyholder's age and gender signifies the policyholder's gender, which is encoded as 0 for female and 1 for male. BMI provides a measure of relative weight to height (kg/m^2), with an ideal BMI value ranging from 18.5 to 25. The number of children of the policyholder. Smoker status identifies whether the policyholder is a smoker or not, encoded as 0 for non-smoker and 1 for smoker. The region denotes the policyholder's region of residence within the United States (US), with categorical values of 0 for northeast, 1 for northwest, 2 for southeast, and 3 for southwest. Cost refers to the medical expenses that individuals incur for health services received. Lastly, the insurance claim class represents the status of insurance claims, with categorical values of 1 for claim and 0 for no claim. This dataset provides valuable information on the various factors that influence health insurance claims.

2.2 Research Steps

The general research workflow is illustrated in Figure 1. The process begins with loading the health insurance dataset into the Google Colaboratory environment. This is followed by the preprocessing stage, which consists of several steps: handling missing values, removing outliers using the interquartile range method, normalizing numerical features using min-max scaling, and performing feature selection based on correlation coefficients to remove irrelevant or highly correlated attributes. After preprocessing, the dataset is split into three subsets, including training data (70%), validation data (10%), and testing data (20%). The training set is used to train the SVM, while the validation set is used to evaluate the performance of the SVM. The test set is reserved for evaluating the final performance of the optimized model.

The SVM is implemented with both linear and non-linear or RBF kernels. To improve the predictive performance of the model, SVM parameters are optimized using the VPPSO algorithm. The VPPSO begins by initializing a population of candidate solutions (particles) that represents possible values of the SVM parameters. Each iteration of the algorithm updates the particle positions based on the best known solutions. At each step, the SVM is trained using the training data, and its performance is evaluated on the validation set to calculate the fitness. If the stopping criteria are not yet met, the process continues by updating the particle velocities and positions. After convergence or early stopping, the best parameter configuration is selected and tested on the test data to obtain the final accuracy, precision, recall, and f_1-score, as well as the computational time. To ensure robustness and avoid randomness in evaluation, the entire VPPSO optimization process is repeated 25 times. The configuration that consistently achieves the best f_1-score and efficient computation is selected as the final model.

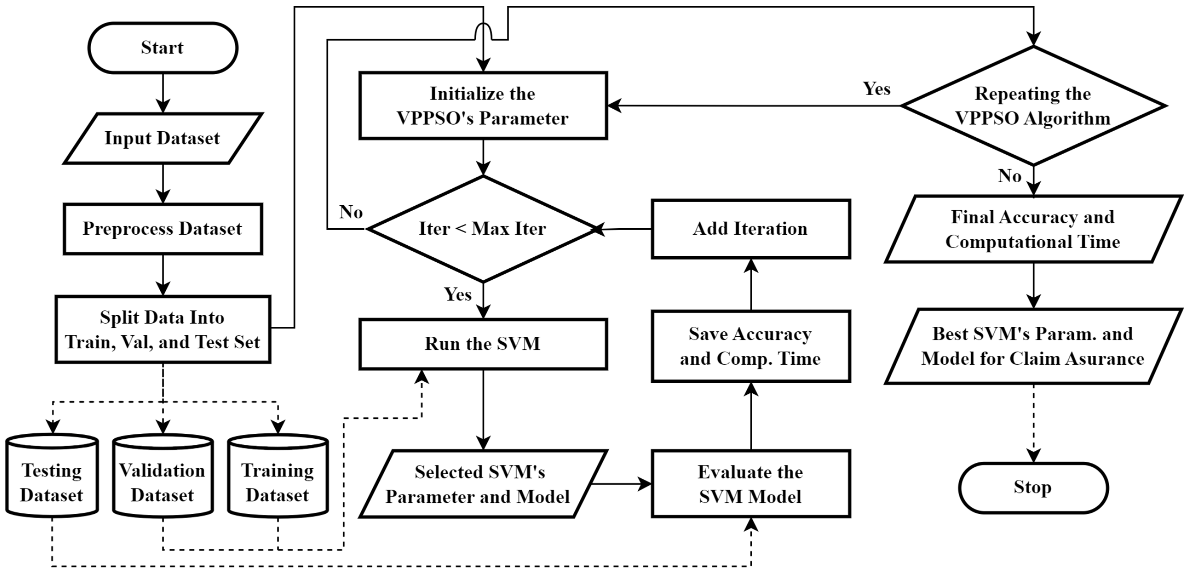

Figure 1: Research Steps

2.3 Support Vector Machine

Support Vector Machine (SVM) was invented by Vapnik in 1992 and has been successful in making predictions in many cases [30]. In classification, SVM operates by finding the optimal hyperplane to separate two classes of data in the feature space [10]. The separated hyperplane can be expressed by the equation $\mathbf{w} \cdot \mathbf{x} + b = 0$, Here, $\mathbf{w}$ is the vector perpendicular to the hyperplane, b is the bias, and $\mathbf{x}$ is a vector variable, $\mathbf{x} = (x^1, x^2, \ldots, x^D)$, with D as the dimensionality of the data [31]. The goal of SVM is to find the best hyperplane that separates data points $\mathbf{x}_i$ based on their labels. The training process of SVM algorithm begins with a random selection of the vector $\mathbf{w}$ and the scalar b. The i-th training data point $\mathbf{x}_i$ must satisfy the following conditions:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq \quad 1 \quad \text{for } y_i = \quad 1, \tag{1}$$
$$\mathbf{x}_i \cdot \mathbf{w} + b < -1 \quad \text{for } y_i = -1, \tag{2}$$

for $i = 1, 2, \dots, n$, where n is the number of training samples. By multiplying both equations (1) and (2) by y_i, the two inequalities become as follows.

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \quad \Rightarrow \quad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \tag{3}$$

with n being the number of training data points.

Suppose that there are two lines H_1 and H_2 that satisfy equations $\mathbf{x}_1 \cdot \mathbf{w} + b = 1$ and $\mathbf{x}_1 \cdot \mathbf{w} + b = -1$, respectively. Let d_1 and d_2 denote the distances from the lines H_1 and H_2 to the separating hyperplane. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the data points closest to the hyperplane, which lie exactly on H_1 and H_2, respectively. The margin distance between the two support vectors with different labels, can be determined by defining a vector $\mathbf{P}$ as the projection of $\mathbf{x}_1 - \mathbf{x}_2$ onto the direction perpendicular to the hyperplane. As such, the following equation holds.

$$\mathbf{P} = (\mathbf{x}_1 - \mathbf{x}_2) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{x}_1 \cdot \mathbf{w} - \mathbf{x}_2 \cdot \mathbf{w}}{\|\mathbf{w}\|} = \frac{(\mathbf{x}_1 \cdot \mathbf{w} + b) - (\mathbf{x}_2 \cdot \mathbf{w} + b)}{\|\mathbf{w}\|}. \tag{4}$$

Since $\mathbf{x}_1$ lies on H_1, then $\mathbf{x}_1 \cdot \mathbf{w} + b = 1$, and since $\mathbf{x}_2$ lies on H_2, then $\mathbf{x}_2 \cdot \mathbf{w} + b = -1$. By substituting these into Equation (4), the equation below is obtained.

$$\mathcal{P} = \frac{1 - (-1)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \tag{5}$$

If it is assumed that the distance between the hyperplane H_1 and H_2 is equal, then the margin is defined $\|\mathbf{w}\|^{-1}$. As such, maximizing the margin is equivalent to minimizing $\|\mathbf{w}\|$ or $2^{-1}\|\mathbf{w}\|^{-1}$. More conveniently, for optimization purposes, the squared norm is used, so the objective is to minimize $\frac{1}{2}\|\mathbf{w}\|^2$. with constraints from Equations (3).

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm was introduced by Kennedy and Eberhart [32]. PSO is inspired by the social behavior of flocking birds and schooling fish. Each particle in PSO navigates with its velocity through the search space to find the optimal solution by updating with their position and velocity. All particles in the swarm are mathematically expressed as follows.

$$\mathbf{v}_j = (v_j^1, v_j^2, v_j^3, \dots, v_j^D), \quad j = 1, 2, \dots, N, \tag{6}$$
$$\mathbf{p}_j = (p_j^1, p_j^2, p_j^3, \dots, p_j^D), \quad j = 1, 2, \dots, N, \tag{7}$$

where $\mathbf{v}_j$ and $\mathbf{p}_j$ denote the velocity and position of the j-th particle, respectively. Here, D represents the number of features or dimensions, and N is the size of the swarm.

At the start of the optimization process, the positions and velocities of all particles are randomly initialized. During each iteration, the particles adjust their velocity and position based on both the global best position found by the swarm (**gbest**) and their personal best position ($\mathbf{pbest}_j$). The update rules are given as follows.

$$\mathbf{v}_j(t+1) = \omega \mathbf{v}_j(t) + c_1 r_1(\mathbf{pbest}_j(t) - \mathbf{p}_j(t)) + c_2 r_2(\mathbf{gbest}(t) - \mathbf{p}_j(t)), \tag{8}$$
$$\mathbf{p}_j(t+1) = \mathbf{p}_j(t) + \mathbf{v}_j(t+1), \tag{9}$$

where t is the current iteration, c_1 and c_2 are the cognitive and social acceleration coefficients, respectively, r_1 and r_2 are random numbers uniformly distributed in the interval $[0, 1]$, and ω is the inertia weight that controls the influence of the previous velocity. The inertia weight ω is typically decreased over time to balance exploration and exploitation. It is computed as follows.

$$\omega = \omega_{\max} - t(\omega_{\max} - \omega_{\min})T^{-1}, \tag{10}$$

where T is the maximum iterations, $\omega_{\max}$ and $\omega_{\min}$ are set to 0.9 and 0.4, respectively [28].

2.5 Velocity Pausing Particle Swarm Optimization (VPPSO)

Velocity pausing is a recent idea in which each particle does not necessarily update its velocity at every iteration [28]. At any given step, a particle can move at the same velocity as in the previous iteration. This idea enables particles to move potentially at three different velocities. These velocities include slower velocity, faster velocity, and constant velocity. Unlike the conventional PSO algorithm, which restricts particles to faster or slower velocities. VPPSO introduces a constant velocity for its particle. This offers a significant advantage by balancing exploitation and exploration, thereby mitigating the risk of premature convergence in standard PSO. For the j-th particle the velocity is updated as follows.

$$\mathbf{v}_j(t+1) = \begin{cases} \mathbf{v}_j(t), & \text{if rand} < \alpha, \\ \omega\,\mathbf{v}_j(t) + c_1\,r_3\big(\mathbf{pbest}_j(t) - \mathbf{p}_j(t)\big) + c_2\,r_4\big(\mathbf{gbest}(t) - \mathbf{p}_j(t)\big), & \text{otherwise,} \end{cases} \tag{12}$$

where $\mathbf{v}_j(t)$ and $\mathbf{v}_j(t{+}1)$ are the velocities in iterations t and $t{+}1$, $\alpha \in (0,1)$ is the velocity-pausing parameter, c_1 and c_2 are the cognitive and social acceleration coefficients, $r_3, r_4 \sim U(0,1)$, and ω is the inertia weight (cf. Eq. (10)). When $\alpha \to 1$ every particle always updates (classic PSO); when $\alpha \to 0$ particles rarely update, sacrificing exploration. Following Shami [28], we set $\alpha = 0.3$.

To improve diversity, the first term of the conventional PSO velocity is replaced, and the explicit inertia term is removed. Therefore, the proposed velocity of VPPSO is as follows.

$$\mathbf{v}_j(t+1) = \mathcal{F}\big(\mathbf{v}_j(t),\, r_5\,a(t)\big) + c_1\,r_6\big(\mathbf{pbest}_j(t) - \mathbf{p}_j(t)\big) + c_2\,r_7\big(\mathbf{gbest}(t) - \mathbf{p}_j(t)\big), \tag{13}$$

with $r_5, r_6, r_7 \sim U(0,1)$ and α is the exponential function that is defined as follows.

$$a(t) = \exp(-(b\,t)^{b} T^{-b}), \quad b = 2.5 \tag{14}$$

The nonlinear operator in Equation (13), that is, $\mathcal{F} : \mathbb{R}^D \times [0,1] \to \mathbb{R}^D$ is defined as follows.

$$\mathcal{F}(\mathbf{v}, k) = \big(|v_1|^k,\, |v_2|^k,\, \ldots,\, |v_D|^k\big), \qquad \mathbf{v} = (v_1, v_2, \ldots, v_D) \in \mathbb{R}^D,\ k \in [0,1]. \tag{15}$$

The update formula for the position of all particles is mathematically expressed as follows.

$$\mathbf{p}_j(t+1) = \mathbf{p}_j(t) + \mathbf{v}_j(t+1). \tag{17}$$

The suggested algorithm divides the total population of particles N into two groups, with the aim of preventing early convergence and preserving diversity. The first group comprises N_1 particles that adjust their positions and velocities using the standard PSO approach, with a modification to the initial term, the velocity of the particle and the application of velocity pausing as outlined in equation (17). The rest of the group consists of N_2 particles. Their positions of these particles are updated exclusively based on the global best solution. Each particle in this second group adjusts its position as follows.

$$\mathbf{p}_j(t+1) = \begin{cases} \mathbf{gbest} + a(t)\,r_8\,\mathcal{F}\big(\mathbf{gbest}, a(t)\big), & r_9 < 0.5, \\ \mathbf{gbest} - a(t)\,r_9\,\mathcal{F}\big(\mathbf{gbest}, a(t)\big), & \text{otherwise,} \end{cases} \tag{18}$$

with $r_8, r_9 \sim U(0,1)$. where r_8 and r_9 are random numbers of uniform distribution within a specified range $[0,1]$. The values of the parameters N_1 and N_2 can be determined as $N_1 = N_2 = [N/2]$ based on the research of Shami [28].

2.6 Parameter Configuration

There are two types of algorithms are used based on the SVM kernel, which are SVM-Linear-VPPSO and SVM-RBF-VPPSO. The SVM-Linear-VPPSO plays a crucial role throughout the model formation process by determining the optimal parameter C value for SVM model. The evaluation process also was conducted using the SVM-RBF-VPPSO, which plays a crucial role by determining the optimal parameter C and γ value for SVM model. Both algorithms aim to achieve the highest f_1-score on the validation data. These algorithms are configured with specific parameters. The maximum iteration is set to 100, and the number of particles varies between 10, 20, and 30. The range for parameter C spans from 0.01 to 3.00. Additionally, the parameter of α is set to 0.3, while the parameters c_1 and c_2 are both assigned a value of 2. These configurations are based on Shami's research [28]. These parameters also guide the velocity update that directs particle movement towards optimal solutions.

The training process is evaluated after 25 iterations. This configuration ensuring that VPPSO halts if there is no significant improvement in the f_1-score. The experiment process is also repeated 25 times to validate robustness and ensure well exploration of the parameter space and reliable result of the VPPSO. This approach aims to enhance classification accuracy for health insurance claim predictions. The termination for SVM-VPPSO is convergence criteria and maximum iterations. The convergence criteria is used when the fitness values of the algorithm do not improve, with a threshold of 10^{-6}.

3 Results and Discussion

This section discusses the results of implementing the SVM model with VPPSO for health insurance claim classification. There are two types of configuration based on the SVM kernel. The results for each configuration with their accuracy, precision, recall, and f_1-score are presented in Table 2 and Table 3.

Based on the results, SVM models without optimization perform worse compared to the SVM-PSO or SVM-VPPSO models. Although the unoptimized SVM model achieved a perfect recall of 100%, it failed to achieve a good precision score. This indicates that while the model successfully identifies all instances of fraudulent or rejected claims, it also misclassifies a significant number of valid claims as fraud. In the context of insurance operations, this problem could lead to unnecessary delays and manual reviews, causing dissatisfaction among legitimate policyholders and increasing operational costs due to unnecessary investigations.

The best performing model configuration is SVM-Linear-VPPSO using 10 particles, which achieved the highest average f_1-score of 90.17% on the testing dataset, as shown in Table 1. In comparison, the best configuration for RBF kernel, that is SVM-RBF-PSO with 20 particles, achieved a slightly lower f_1-score of 89.78% as shown in Table 3. This confirms that the linear kernel is suitable for this insurance claim dataset. Moreover, the minimal difference of just 0.19% between training and testing f_1-scores for SVM-Linear-VPPSO suggests that the model generalizes well and does not overfit, making this congifuration more reliable for deployment in real world environments.

Further analysis of the result shows that the SVM-Linear-VPPSO with 10 particles achieved 88.27% accuracy, 93.19% precision, and 87.35% recall on the testing dataset. High precision score indicates that most claims predicted as fraud are indeed fraudulent, minimizing the risk of wrongly flagging genuine claims. The balance between precision and recall ensures that insurers can automate claim review processes more confidently, without sacrificing customer trust.

Optimization of the SVM model regularization parameter (C) using VPPSO resulted in effective parameter values with $C = 0.8123$, $C = 1.002$, and $C = 0.9455$ using 10, 20, and 30 particles, respectively, allowing better model fitting. Notably, the linear kernel also required less computational time than the RBF kernel, reinforcing its practicality.

Table 2: Evaluation Result of SVM-Linear-VPPSO

Evaluation Metric		SVM with C = 0.01	SVM with C = 0.10	SVM PSO 10 par.	SVM VPPSO 10 par.	SVM PSO 20 par.	SVM VPPSO 20 par.	SVM PSO 30 par.	SVM VPPSO 30 par.
F_1-score	Train	73.39%	89.88%	90.04%	89.98%	90.03%	90.00%	90.07%	90.00%
	Valid	74.82%	89.70%	88.89%	88.77%	88.95%	88.80%	**89.12%**	88.86%
	Test	76.21%	89.49%	90.04%	**90.17%**	90.04%	90.16%	89.82%	90.06%
Recall	Train	**100.00%**	92.62%	89.84%	89.48%	90.10%	89.55%	90.68%	89.65%
	Val	**100.00%**	91.46%	88.42%	88.21%	88.53%	88.26%	**88.84%**	88.37%
	Test	**100.00%**	90.30%	87.54%	87.35%	87.93%	87.47%	88.24%	87.54%
Precision	Train	57.97%	87.30%	90.26%	90.51%	90.00%	90.47%	89.50%	90.37%
	Valid	59.77%	88.00%	89.36%	89.34%	89.37%	89.34%	**89.41%**	89.36%
	Test	61.57%	88.69%	92.73%	**93.19%**	92.29%	93.03%	91.49%	92.76%
Accuracy	Train	58.01%	87.93%	**88.49%**	88.47%	88.45%	88.48%	88.42%	88.46%
	Valid	59.79%	87.44%	87.46%	87.34%	87.52%	87.37%	**87.70%**	87.43%
	Test	61.57%	86.94%	88.07%	**88.27%**	88.01%	88.24%	87.67%	88.10%

Table 3: Evaluation Result of SVM-RBF-VPPSO

Evaluation Metric		SVM C = 0.01 $\gamma = 1.00$	SVM C = 0.10 $\gamma = 0.01$	SVM PSO 10 par.	SVM VPPSO 10 par.	SVM PSO 20 par.	SVM VPPSO 20 par.	SVM PSO 30 par.	SVM VPPSO 30 par.
F_1-score	Train	73.34%	73.34%	94.12%	94.07%	93.83%	94.05%	94.06%	**94.15%**
	Valid	74.79%	74.79%	93.44%	93.26%	93.50%	93.54%	**93.63%**	93.58%
	Test	76.21%	76.21%	89.77%	89.72%	**89.78%**	89.60%	89.77%	89.53%
Recall	Train	**100.00%**	**100.00%**	92.72%	92.63%	92.44%	92.61%	92.66%	92.70%
	Val	**100.00%**	**100.00%**	91.42%	90.95%	91.21%	91.00%	91.21%	90.95%
	Test	**100.00%**	**100.00%**	86.38%	86.35%	86.18%	85.89%	86.16%	85.75%
Precision	Train	57.91%	57.91%	95.55%	95.57%	95.25%	95.54%	95.49%	**95.65%**
	Valid	59.74%	59.74%	95.57%	95.69%	95.92%	96.23%	96.19%	**96.39%**
	Test	61.57%	61.57%	93.45%	93.37%	93.70%	93.66%	**93.71%**	93.68%
Accuracy	Train	57.91%	57.91%	93.29%	93.24%	92.95%	93.22%	93.22%	**93.33%**
	Valid	59.74%	59.74%	92.72%	92.54%	92.81%	92.87%	**92.96%**	92.93%
	Test	61.57%	61.57%	87.88%	87.82%	**87.93%**	87.73%	87.91%	87.66%

In terms of computational efficiency, as shown in Table 4, unoptimized SVM had the shortest runtime, that is 0.048–0.114 seconds, but its performance was inadequate for accurate insurance claim classification. While SVM-PSO model had lower computation time than SVM-VPPSO, it produced slightly inferior prediction results. The best trade-off was achieved by the SVM-Linear-VPPSO with 10 particles, which delivered the highest f_1-score of 90.17% in just 46.938 seconds for 25 iterations. This result is three times faster than the 20-particle configuration. This makes it efficient and accurate for practical use.

This research investigated the performance variability of different combinations of particle number, as shown in Table 5 and 6. The experiment results indicate very small standard deviations. It suggests that the experimental outcomes exhibit low variation around the mean value, demonstrating that the model delivers consistent performance. Small standard deviations also provide more accurate and reliable estimates of model performance metrics such as accuracy, precision, recall, and f_1-score. A reliable and consistent model is necessary because inconsistency in the evaluation of claims can lead to legal disputes, poor insurer experience, and compliance issues.

Based on the result in Tables 5 and 6, the SVM-Linear-VPPSO method with 10 particles achieved smallest standard deviation of 0.24%, indicating robustness. Furthermore, in RBF kernel, the SVM-RBF-PSO using 30 particles achieved the smallest standard deviation of 0.41%. Based on this comparison, the SVM-Linear-VPPSO method with 10 particles is identified as the best method, because it produces a highly accurate, stable, and computationally efficient model

Table 4: Time Computation Result of Each Configuration

SVM's Parameters	Kernel	Opt.	Num. of Particle	Mean of Time Comp.	Std of Time Comp.
–	Linear	VPPSO	10	46.938s	1.832s
–	Linear	VPPSO	20	137.193s	37.909s
–	Linear	VPPSO	30	333.362s	116.409s
–	Linear	PSO	10	151.609s	27.087s
–	Linear	PSO	20	531.940s	177.370s
–	Linear	PSO	30	1533.447s	481.073s
C = 0.10	Linear	–	–	0.048s	0.003s
C = 0.01	Linear	–	–	0.060s	0.005s
–	RBF	VPPSO	10	96.427s	44.294s
–	RBF	VPPSO	20	293.307s	108.715s
–	RBF	VPPSO	30	607.674s	246.947s
–	RBF	PSO	10	147.329s	30.643s
–	RBF	PSO	20	767.822s	279.229s
–	RBF	PSO	30	1915.929s	458.981s
C = 0.10, $\gamma = 0.01$	RBF	–	–	0.098s	0.006s
C = 0.01, $\gamma = 1.00$	RBF	–	–	0.114s	0.008s

Table 5: Standard Deviation Result of SVM-Linear-VPPSO

Evaluation Metric		SVM PSO 10 par.	SVM VPPSO 10 par.	SVM PSO 20 par.	SVM VPPSO 20 par.	SVM PSO 30 par.	SVM VPPSO 30 par.
F$_1$-score	Train	0.11%	0.12%	0.16%	0.13%	0.11%	0.13%
	Valid	0.11%	0.15%	0.16%	0.20%	0.11%	0.20%
	Test	0.30%	0.24%	0.34%	0.27%	0.37%	0.27%
Recall	Train	0.37%	0.75%	0.40%	0.80%	0.41%	0.80%
	Valid	1.09%	0.26%	1.34%	0.36%	1.38%	0.36%
	Test	0.54%	0.64%	0.60%	0.67%	0.67%	0.67%
Precision	Train	0.82%	0.58%	1.03%	0.65%	1.09%	0.65%
	Valid	0.95%	0.03%	1.06%	0.04%	1.18%	0.04%
	Test	0.06%	0.92%	0.06%	1.08%	0.07%	1.08%
Accuracy	Train	1.58%	0.10%	1.68%	0.13%	1.88%	0.13%
	Valid	0.12%	0.15%	0.12%	0.21%	0.10%	0.21%
	Test	0.30%	0.34%	0.34%	0.40%	0.38%	0.40%

for health insurance claim prediction. The proposed SVM-Linear-VPPSO model enables insurers to automate the claims classification more reliably, reducing the fraud risk while maintaining customer satisfaction and minimizing operational delays. This makes the model highly suitable for real-world deployment in health or general insurance claim management systems.

Despite the results, several limitations of this study must be acknowledged. First, the evaluation is limited to the SVM classifier and its optimization using PSO and VPPSO; other advanced classification algorithms, such as RF, XGBoost, or Deep Neural Networks were not included in the comparison. This restricts the broader generalization of the model's superiority. Second, the dataset used in this research is sourced from a single dataset and not fully represent the complexity, imbalance, or evolving patterns found in real-world insurance claims. Third, while VPPSO demonstrated consistent results across multiple runs, but the optimization process still relies on static particle configurations and fixed maximum iteration limits, which may not adapt well to complex or larger-scale datasets. Lastly, the model does not incorporate temporal or sequential aspects of claim data, such as trends in claim history or time-series behaviors, which are often relevant in practical fraud detection. These limitations present opportunities for future research to explore more diverse ML models, utilize real-world multi-source datasets, and develop adaptive or dynamic optimization frameworks that can handle temporal patterns and data complexity more effectively.

Table 6: Standard Deviation Result of SVM-RBF-VPPSO

Evaluation Metric		SVM PSO 10 par.	SVM VPPSO 10 par.	SVM PSO 20 par.	SVM VPPSO 20 par.	SVM PSO 30 par.	SVM VPPSO 30 par.
F_1-score	Train	**0.55%**	0.74%	0.73%	**0.55%**	0.56%	**0.55%**
	Valid	0.35%	0.36%	**0.29%**	0.36%	**0.29%**	0.36%
	Test	0.42%	0.60%	0.53%	0.44%	**0.41%**	0.44%
Recall	Train	0.58%	0.58%	0.64%	**0.55%**	**0.52%**	0.55%
	Valid	0.67%	**0.44%**	0.63%	0.49%	0.63%	0.49%
	Test	0.99%	1.07%	1.11%	**0.93%**	0.97%	**0.93%**
Precision	Train	0.61%	0.95%	0.90%	**0.59%**	0.62%	**0.59%**
	Valid	1.25%	**0.87%**	1.09%	1.04%	1.20%	1.04%
	Test	0.70%	0.58%	**0.36%**	0.50%	0.51%	0.50%
Accuracy	Train	0.63%	0.86%	0.85%	**0.63%**	0.65%	**0.63%**
	Valid	0.45%	0.43%	**0.37%**	0.44%	0.38%	0.44%
	Test	0.46%	0.66%	0.56%	0.47%	**0.43%**	0.47%

4 Conclusion

This study introduces a novel implementation of the VPPSO algorithm to enhance SVM performance in the classification of health insurance claims. The experimental results confirm that the SVM-Linear-VPPSO configuration with 10 particles achieves the best balance of accuracy and efficiency, producing the highest f_1 score of 90. 17%, accuracy of 88. 27%, precision of 93. 19% and recall of 87. 35% in the test dataset. It also shows no signs of overfitting, evidenced by a 0.19% f_1-score differences between training and testing data. Additionally, the configuration demonstrates efficiency with a computation time of only 46.938 seconds, and a low time variance of 1.832 seconds. The proposed SVM-VPPSO also outperforms the SVM and SVM-PSO configurations, not only in predictive accuracy but also in computational robustness and stability, as shown in the small standard deviation of 0.24% across 25 repeated runs. These suggest that the VPPSO mechanism effectively mitigates premature convergence and offers reliable parameter optimization for SVM in real-world insurance datasets.

However, this study only focused on SVM and did not compare them with other ML classifiers such as RF, XGBoost, or ANN. The dataset is limited to a single source and does not capture all of the variability of real-world insurance data. Future research should explore broader classifier model comparisons, incorporate with multi-source and time-dependent insurance data, and investigate more optimization metaheuristic algorithm to find the best configuration model to predict claim classification. These would advance automated, accurate, and scalable insurance claim classification systems.

CRediT Authorship Contribution Statement

Luh Putu Dharma Jayanti: Conceptualization, Methodology, Software, Investigation, Writing–Original Draft. **Syaiful Anam:** Validation, Writing–Review & Editing, Supervision, Project Administration. **Safrizal Ardana Ardiyansa:** Investigation, Resources,Writing–Review & Editing. **Natasha Clarissa Maharani:** Investigation, Writing–Review & Editing.

Declaration of Generative AI and AI-assisted technologies

During the preparation of this manuscript, AI-assisted technologies were utilized for writing assistance and grammar checking. Specifically, Gemini and ChatGPT were employed as writing assistants to assist in drafting and refining sections of the text. Grammarly was used for grammar and spelling proofreading to improve the linguistic quality of the manuscript.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest regarding the publication of this article. This includes both financial and non-financial interests that could be perceived to influence the research outcomes. The study was conducted entirely independently of commercial or proprietary relationships, ensuring the integrity and objectivity of the findings.

Funding and Acknowledgments

This research did not receive external funding. The authors acknowledge the support and facilities provided by Brawijaya University, which were instrumental in the completion of this work.

Data and Code Availability

The dataset utilized in this study is publicly available on Kaggle at `https://www.kaggle.com/datasets/easonlai/sample-insurance-claim-prediction-dataset`. The code developed and used for data preprocessing, feature selection, and model implementation is available from the corresponding author upon reasonable request.

References

[1] C. C. Selisteanu, "The Insurance Contract - A Support In The Business Sphere," *Perspectives of Law and Public Administration*, vol. 5, no. 1, pp. 161–169, Nov. 2016. Available online.

[2] N. G., B. A. V. Rao, and P. Kengnal, "Utilization, satisfaction, out of pocket expenditure and health seeking behaviour among the insured residents of rural field area: A cross sectional study," *International Journal Of Community Medicine And Public Health*, vol. 7, no. 3, pp. 1047–1050, Feb. 2020. DOI: `10.18203/2394-6040.ijcmph20200964`. Available online.

[3] M. M. Billah, *Islamic Insurance Products*. Cham: Springer International Publishing, 2019. DOI: `10.1007/978-3-030-17681-5`. Available online.

[4] Nabila Erinaputri, Rhaina Al Yasin, Shifa Aulia Maghfiroh, and Anisya Febriyanti, "The Level Of Community's Sense Of Importance In Ownership Of Health Insurance," *Jurnal Ilmu Kedokteran dan Kesehatan Indonesia*, vol. 3, no. 2, pp. 01–11, May 2023. DOI: `10.55606/jikki.v3i2.1531`. Available online.

[5] Otoritas Jasa Keuangan, "Perasuransian," in *Seri literasi keuangan perguruan tinggi*, Otoritas Jasa Keuangan, Ed., Jakarta: Otoritas Jasa Keuangan, 2019, ch. 6. DOI: `https://sikapiuangmu.ojk.go.id/FrontEnd/LiterasiPerguruanTinggi/`. Available online.

[6] Z. Rouaine, M. Jerry, and Q. Ahlam, "Automobile insurance: Analysis of the impact of a premium change on the behavior of insured at the time of subscription and termination," *Journal of Business and Economics*, vol. 10, pp. 517–530, Jun. 2019. DOI: `10.15341/jbe(2155-7950)/06.10.2019/003`.

[7] S. H. Rambe, S. Hamidah, and S. Wicaksono, "Perlindungan hukum bagi tertanggung atas gagal klaim asuransi kesehatan akibat ketidaklengkapan data pemeriksaan kesehatan," id, *Brawijaya Law Student Journal*, Jan. 2020. Available online.

[8] S. H. Rambe and P. Sekarayu, "Perlindungan Hukum Nasabah Atas Gagal Klaim Asuransi Akibat Ketidaktransparanan Informasi Polis Asuransi," *JURNAL USM LAW REVIEW*, vol. 5, no. 1, pp. 93–109, Apr. 2022. DOI: `10.26623/julr.v5i1.4073`. Available online.

[9] M. Batubara, P. R. Silalahi, R. Fachrina, I. A. Putri, and F. Prayogi, "Analisis Kasus Gagal Bayar Klaim Nasabah dalam Perusahaan Asuransi Jiwasraya," *Jurnal Kajian Ekonomi dan Bisnis Islam*, vol. 3, no. 4, pp. 633–640, 2022. DOI: 1047467/elmal.v3i4.989.

[10] J. Han, M. Kamber, and J. Pei, "8 - classification: Basic concepts," in *Data Mining: Concepts and Techniques (Third Edition)*, ser. The Morgan Kaufmann Series in Data Management Systems, J. Han, M. Kamber, and J. Pei, Eds., Third Edition, Boston: Morgan Kaufmann, 2012, pp. 327–391. DOI: https://doi.org/10.1016/B978-0-12-381479-1.00008-3. Available online.

[11] A. M. Walker, "Pattern recognition in health insurance claims databases," *Pharmacoepidemiology and Drug Safety*, vol. 10, no. 5, pp. 393–397, 2001. DOI: https://doi.org/10.1002/pds.611. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pds.611. Available online.

[12] A. Harale, Y. Dubey, V. Gupta, A. Motghare, M. Chakole, and A. Pathade, "Empirical analysis of predictive models for insurance claim classification," in *2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS)*, 2022, pp. 333–336. DOI: 10.1109/ICETEMS56252.2022.10093335.

[13] A. Goh and S.-H. Goh, "Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data," *Computers and Geotechnics*, vol. 34, Aug. 2007. DOI: 10.1016/j.compgeo.2007.06.001.

[14] O. Devos, C. Ruckebusch, A. Durand, L. Duponchel, and J.-P. Huvenne, "Support vector machines (svm) in near infrared (nir) spectroscopy: Focus on parameters optimization and model interpretation," *Chemometrics and Intelligent Laboratory Systems*, vol. 96, pp. 27–33, Mar. 2009. DOI: 10.1016/j.chemolab.2008.11.005.

[15] R. Safdari, P. Rezaei, M. Ghazisaeedi, and T. Samad-Soltani, "Evaluation of classification algorithms vs knowledge-based methods for differential diagnosis of asthma in iranian patients," *International Journal of Information Systems in the Service Sector*, vol. 10, p. 14, Apr. 2018. DOI: 10.4018/IJISSS.2018040102.

[16] G. Liu, D. Zhou, H. Xu, and C. Mei, "Model optimization of svm for a fermentation soft sensor," *Expert Systems with Applications*, vol. 37, pp. 2708–2713, Apr. 2010. DOI: 10.1016/j.eswa.2009.08.008.

[17] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine teori dan aplikasinya," in *Proceeding of Indonesian Scientific Meeting in Central Japan*, IlmuKomputer.Com, 2003.

[18] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *The Journal of Machine Learning Research*, vol. 2, Jan. 2002. DOI: 10.1162/153244302760185252.

[19] Styawati, Andi Nurkholis, Zaenal Abidin, and Heni Sulistiani, "Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly Pada Data Opini Film," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 904–910, Oct. 2021. DOI: 10.29207/resti.v5i5.3380. Available online.

[20] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of pso algorithm," *Applied Sciences*, vol. 12, no. 17, 2022. DOI: 10.3390/app12178392. Available online.

[21] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of pid controller in avr system," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384–391, 2004. DOI: 10.1109/TEC.2003.821821.

[22] S. Lin, K. Ying, S. Chen, and Z. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817–1824, Nov. 2008. DOI: 10.1016/j.eswa.2007.08.088.

[23] S. Anam, M. R. Putra, Z. Fitriah, I. Yanti, N. Hidayat, and D. Mahanani, "Health claim insurance prediction using support vector machine with particle swarm optimization," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 2, pp. 0797–0806, Jun. 2023. DOI: 10.30598/barekengvol17iss2pp0797-0806. Available online.

[24] B. Jana, S. Mitra, and S. Acharyaa, "Repository and mutation based particle swarm optimization (rmpso): A new pso variant applied to reconstruction of gene regulatory network," *Applied Soft Computing*, Oct. 2018. DOI: 10.1016/j.asoc.2018.09.027.

[25] Z. Zhou, F. Li, J. H. Abawajy, and C. Gao, "Improved pso algorithm integrated with opposition-based learning and tentative perception in networked data centres," *IEEE Access*, vol. 8, pp. 55 872–55 880, 2020. DOI: 10.1109/ACCESS.2020.2981972.

[26] S. Ardiyansa, N. Maharani, S. Anam, and E. Julianto, "Optimizing Heart Attack Diagnosis Using Random Forest with Bat Algorithm and Greedy Crossover Technique," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 18, no. 2, pp. 1053–1066, May 2024. DOI: 10.30598/barekengvol18iss2pp1053-1066. Available online.

[27] O. Muhammed Neda and A. Ma'arif, "Chaotic Particle Swarm Optimization for Solving Reactive Power Optimization Problem," *International Journal of Robotics and Control Systems*, vol. 1, no. 4, pp. 523–533, Jan. 2022. DOI: 10.31763/ijrcs.v1i4.539. Available online.

[28] T. M. Shami, S. Mirjalili, Y. Al-Eryani, K. Daoudi, S. Izadi, and L. Abualigah, "Velocity pausing particle swarm optimization: A novel variant for global optimization," *Neural Computing and Applications*, vol. 35, Jan. 2023. DOI: 10.1007/s00521-022-08179-0.

[29] K. Tang and C. Meng, "Particle swarm optimization algorithm using velocity pausing and adaptive strategy," *Symmetry*, vol. 16, no. 6, 2024. DOI: 10.3390/sym16060661. Available online.

[30] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifier," *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, vol. 5, Aug. 1996. DOI: 10.1145/130385.130401.

[31] Z. Jun, "The Development and Application of Support Vector Machine," *Journal of Physics: Conference Series*, vol. 1748, no. 5, p. 052 006, Jan. 2021. DOI: 10.1088/1742-6596/1748/5/052006. Available online.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.