Computing (2025) 107:66
https://doi.org/10.1007/s00607-025-01425-y

REGULAR PAPER

Hybrid feature selection module for improving performance of software vulnerability severity prediction model on textual dataset

Ruchika Malhotra[1] · **Vidushi**[1,2]

Received: 13 September 2023 / Accepted: 20 January 2025 / Published online: 6 February 2025

Abstract

Software vulnerability severity prediction is a critical area in software engineering, where model performance heavily depends on the quality of the feature set used for training. Challenges such as feature redundancy, correlations, and irrelevant features can degrade model effectiveness, emphasizing the importance of Feature Selection (FS) methods to optimize performance and reduce development costs. In this study, we introduce two innovative FS modules within the homogeneous wrapper method category. The first, Parallel-Grey Wolf Optimization (P-GWO), employs a hybrid approach combining Grey Wolf Optimization (GWO) with Opposition-Based Learning (OBL). The second, Multi-Stage Grey Wolf Whale Optimization (MS-G2WO), uses GWO to find an initial optimal solution, which is further refined by the Whale Optimization Algorithm (WOA). Both modules are evaluated using Area Under Curve (AUC) values, demonstrating the significant impact of FS on model performance. Our experimental results show that P-GWO achieved superior performance with a mean AUC of 0.804, followed by MS-G2WO with a mean AUC of 0.77, establishing the effectiveness of these proposed methods in improving vulnerability severity prediction models.

Keywords Software vulnerability · Severity prediction · Prediction model · Feature selection

✉ Vidushi
vidushi_2k18phdco23@dtu.ac.in

[1] Software Engineering Department, Delhi Technological University, Delhi, India

[2] School of Engineering & Technology, Vivekananda Institute of Professional Studies - Technical Campus, Delhi, India

1 Introduction

With advancements in technology, the demand for software systems has grown significantly. Large and complex software systems are being developed to meet customer requirements, making it increasingly difficult to create completely error-free software without vulnerabilities. As a result, ensuring the security of software systems has become a top priority for developers [1–3]. The number of reported software vulnerabilities is rising exponentially [4, 5], and once a vulnerability is reported, the likelihood of it being exploited increases dramatically [6]. Therefore, there is a need to develop a model that can automatically predict the severity of vulnerabilities, allowing developers to prioritize the more critical vulnerabilities for faster resolution. Such a model can be referred to as a Software Vulnerability Severity Prediction Model (SVSPM).

Several vulnerability scoring frameworks have been proposed for assessing vulnerabilities [7–9]. Among these, the Common Vulnerability Scoring System (CVSS) [10], developed by the Forum of Incident Response and Security Teams (FIRST), is recognized as a standard for evaluating the severity of software vulnerabilities. Once a vulnerability is identified, it is recorded in the Common Vulnerabilities and Exposures (CVE) database [11], which is an example of a vulnerability database. This database contains vulnerability reports in textual format. When this textual data is input into the SVSPM, it generates high-dimensional feature vectors.

Therefore, a need arises to apply Feature Selection (FS), to reduce the dimension of feature vectors. This is done by removing noisy, redundant, and irrelevant features. This will aid in the improvement in the prediction capability of the prediction model [12, 13].

FS approaches can be divided into two categories, filter-based approach and wrapper-based approach [14]. In the filter approach, features are selected based on their relevance to the target variable across the entire dataset. This is achieved by evaluating each feature independently using statistical measures such as correlation, mutual information, or other ranking criteria. Features that meet a predefined threshold are retained, while others are filtered out. While the filter approach is computationally fast, it can be considered a brute-force method, as it evaluates each feature independently without considering interactions between features. As a result, it may not yield the best classification results since it operates independently of any specific learning algorithm. The wrapper approach, in contrast, involves two key components: search algorithms and classifiers. Search algorithms systematically explore different feature subsets using methods like metaheuristic nature-inspired algorithms which apply evolutionary strategies to refine subsets. Classifiers evaluate the predictive performance of each subset by training a model on it, helping identify the best feature set for optimal model performance, making the wrapper approach perform better than the filter approach [15]. To find the optimal set of features from a high-dimensional feature space using a brute-force approach is not feasible. For instance, if a dataset has n features, then the total number of solutions will be 2^n-1. Therefore, the problem of

FS is NP-hard and has exponential computational complexity [16]. Therefore, because of these reasons, metaheuristic nature-inspired algorithms have gained popularity in solving many real-time optimization problems, producing good results in less time [17–19]. Genetic Algorithm (GA) [20], Particle Swarm Optimization (PSO) [21], Grey Wolf Optimization (GWO) [22], and Whale Optimization Algorithm (WOA) [23] are a few metaheuristic algorithms. Recent advancements in metaheuristic algorithms, such as the Liver Cancer Algorithm (LCA), Parrot Optimizer (PO), Artemisinin Optimization (AO), Moth Search Algorithm (MSA), Fata Morgana Algorithm (FATA), and Polar Lights Optimization (PLO), offer innovative approaches to solving optimization problems. Compared to these, Grey Wolf Optimization (GWO) and Whale Optimization Algorithm (WOA) stand out for their simpler frameworks and lower computational overhead. While LCA, PO, and FATA focus heavily on exploration, GWO and WOA maintain a more balanced approach. Additionally, WOA's spiral and bubble-net strategies provide a unique edge in convergence, rivaling the precision of AO and PLO, while GWO's hierarchical structure offers robust global search capabilities akin to MSA. These comparisons highlight the versatility of GWO and WOA in tackling diverse optimization challenges.

Many metaheuristic algorithms have problems like reaching to local optimum early, lack of diversity in searching, and an imbalance among explorative and exploitive phases of the algorithm. These limitations can negatively impact their ability to find optimal solutions, particularly in high-dimensional feature spaces. Additionally, most FS techniques focus on optimizing a single objective, such as accuracy or relevance. Such single-objective optimization may not adequately address the multifaceted requirements of modern ML tasks. Moreover, traditional FS methods struggle to handle multi-label datasets or integrate multiple data views, which are increasingly common in complex datasets. This highlights the need for FS approaches that balance multiple criteria and adapt to the diverse characteristics of datasets. All these problems motivated researchers to create hybrid modules, which combine two or more approaches for FS aiming at integrating the strength of individual FS approaches and overpowering their weaknesses [24].

Therefore, in this paper, we created hybrid modules, which combine two or more approaches for FS aiming at integrating the strength of individual FS approaches and overpowering their weaknesses [24]. A fitness function combining error rate in prediction and size of feature subset is used to evaluate solutions, promoting a balance between predictive accuracy and minimal feature usage. The modules were tested on vulnerability datasets sourced from the CVE database, which is a multi-label dataset. These two FS modules are based on metaheuristic algorithms. Due to their proven effectiveness, simplicity, and balance of exploration and exploitation, we choose to work with the GWO and WOA algorithms. GWO is a nature-inspired metaheuristic algorithm that mimics the leadership hierarchy and hunting behavior of grey wolves to explore and exploit the search space effectively. WOA is a metaheuristic algorithm inspired by the bubble-net hunting strategy of humpback whales, balancing exploration and exploitation in optimization tasks. GWO and WOA are effective for feature selection due to their robust exploration-exploitation balance, global search capabilities, and minimal

parameter dependencies. Both GWO's leadership hierarchy and WOA's bubble-net foraging provide robust global and local search capabilities, which are essential for feature selection. Additionally, GWO and WOA share compatible convergence mechanisms and minimal parameter dependencies, making them ideal for hybridization. This allows the hybrid algorithm to first explore the search space broadly with GWO, and then fine-tune feature selection using WOA, achieving better optimization without premature convergence. Recent studies have also demonstrated the successful hybridization of GWO and WOA in various optimization problems, showing improvements in convergence speed, solution quality, and the ability to handle complex search spaces [25–27]. Firstly, a Parallel- Grey Wolf Optimization (P-GWO) module is designed in which the OBL approach is used to generate the opposite solution to the initial solution generated by GWO and both these solutions are hybridized aiming to solve the issue of reaching to local optimum early. Opposition-Based Learning (OBL) is a computational learning paradigm inspired by the concept of opposites. It leverages the idea that considering both a solution and its opposite can improve the efficiency and performance of algorithms. OBL is commonly used in optimization, machine learning, and artificial intelligence to enhance convergence speed, exploration, and decision-making. By evaluating both the current candidate and its opposite, the approach aims to find optimal solutions more effectively. Secondly, a Multi-Stage Grey Wolf Whale Optimization (MS-G2WO) module is designed in which the first algorithm finds the best solution and reduces the search space for the second algorithm, increasing the convergence speed and improving the best-found solution. The choice of GWO and WOA is made as both are recently introduced metaheuristic algorithms gaining popularity in the field of FS [28–33] and work on a common principle of hunting behavior. Their common principle of working made it convenient for us to integrate and hybridize the two algorithms. Thus, the primary objective of these modules is to enhance the population's quality, leading to an improved optimal solution.

The contribution of this paper can be highlighted in the points mentioned below:

1. Two innovative hybrid modules are introduced for feature selection (FS): the P-GWO module and the MS-G2WO module.
2. The proposed modules, P-GWO and MS-G2WO, are evaluated against native algorithms and various metaheuristic algorithms. Experimental results demonstrate that the proposed modules achieve superior performance in FS.
3. The effectiveness of the P-GWO and MS-G2WO modules is further
4. The paper introduces a novel integration framework that combines the strengths of different optimization strategies, paving the way for future advancements in hybrid feature selection methodologies.

The Research Questions (RQs) addressed in the study are as follows:

RQ1. What is the improvement in the performance of SVSPM when proposed FS modules are applied to reduce the dimension of the data?

The effectiveness of various methods used for dimensionality reduction is measured on the dataset of five different products of Mozilla using Area Under Curve (AUC) as a performance measure.

RQ1.1. What is the improvement in the performance of SVSPM when the results of proposed modules P-GWO and MS-G2WO are compared when all features are used and no FS is done?

The result values are compared dataset-wise and improvement in the performance is recorded. The comparison is made between the models developed using the proposed module for FS and the model developed using no FS.

RQ1.2. What is the improvement in the performance of SVSPM when the result comparison of proposed modules P-GWO and MS-G2WO is made with state-of-the-art algorithms?

Similar to RQ1.1, the comparison is made between the models developed using the proposed module for FS and the model developed using PSO and GA.

RQ1.3. What is the improvement in the performance of SVSPM when the results of proposed modules P-GWO and MS-G2WO are compared with native algorithms?

Similar to RQ1.1, the comparison is made between the models developed using the proposed modules for FS and the model developed using the native algorithms.

RQ2. Which module of FS among the two proposed modules gave better results?

Addressing this RQ, the most effective method of FS is found, and a final judgment is given, stating the best method for FS.

RQ3. Whether the performance of SVSPM developed boosted significantly with the help of FS approaches?

Friedman test and Wilcoxon signed-rank test are applied and a significant difference in the performance of SVSPM is found.

The structure of the paper is as follows: Section 2 describes the work done in the literature related to this area. Section 3 lists all the preliminaries used to carry out the experiment of the paper. Section 4 describes the proposed methodology. Section 5 analyzes the RQs and determines the performance of the proposed methodology. Section 6 provides various threats to validity that are present in the research work done in the paper. Section 7 lays out a summary of the work done, states a conclusion about the current work, and some points regarding future work.

2 Related work

The SVSPM prediction model is primarily built using ML algorithms. To enhance performance, researchers integrated FS techniques with specific classifiers. In the following subsections, we will review previous studies on SVSPM development and explore research on FS employing hybrid metaheuristic algorithms.

3 Software vulnerability severity prediction

Software vulnerability prediction models rely heavily on machine learning (ML) algorithms. To enhance performance, researchers often integrate FS techniques with classifiers. Below, we summarize key works and critically analyze their contributions.

Filter-Based FS Approaches: Gupta et al. [34] employed Information Gain (IG) for FS, combining it with classifiers such as Naïve Bayes (NB), JRIP, J48, Random Tree, and Bagging, with the J48 classifier achieving an average true rate of 87.8%. Huang et al. [35] utilized a Deep Neural Network (DNN) with IG, outperforming Support Vector Machine (SVM), NB, and KNN in accuracy (0.87), recall (0.82), precision (0.85), and F1-score (0.81). Similarly, Chen et al. [36] combined IG and Chi-Square (CHI-2) with advanced term weighting techniques like Term Frequency-Inverse Gravity Moment (TF-IGM), demonstrating superiority over Term Frequency-Inverse Document Frequency (TF-IDF) across multiple ML algorithms. Malhotra et al. [37] found that NB coupled with IG provided the best performance metrics among other methods.

Wrapper-Based FS Approaches: Han et al. [38] implemented word embeddings with a shallow Convolutional Neural Network (CNN), achieving f-measure scores ranging from 0.751 to 0.882 across severity levels. Dam et al. [39] used Long Short-Term Memory (LSTM) models to capture semantic and syntactic features, with precision, recall, and f-measure values of 0.92, 0.93, and 0.91, respectively. Russell et al. [40] highlighted the efficacy of deep feature representation with CNN + Random Forest (RF), achieving an AUC of 0.904. Ban et al. [41] employed Bi-directional LSTM (Bi-LSTM) for feature representation, testing classifiers like J48, KNN, Linear Discriminant Analysis (LDA), Neural Networks (NN), RF, and SVM. Gong et al. [42] explored neural network models for multi-target classification, and Lin et al. [43] and Liu et al. [44] applied deep learning frameworks incorporating LSTM.

While filter-based FS approaches demonstrate computational efficiency and memory use, their performance is often dataset-specific. On the other hand, wrapper-based methods like CNN and LSTM excel in predictive accuracy but suffer from higher computational costs. Deep learning frameworks dominate in scalability and accuracy but require extensive tuning. These studies lack extensive exploration of parallelization strategies and multi-stage hybridization, which this paper addresses with a novel multi-stage hybrid approach. Zhou et al. [45] propose an unsupervised feature selection method for balanced clustering keeping in mind the intrinsic balanced structure of data. Jiang et al. [46] propose a fine-grained feature-based approach for Android malware detection, combining static and dynamic analysis to enhance accuracy. Their framework extracts nuanced app behaviors and employs machine learning for effective malware identification. Zhang et al. [47] propose a robust feature learning method for adversarial defense through hierarchical feature alignment, enhancing model resilience against adversarial attacks.

4 Feature selection using hybrid metaheuristic approach

FS, an NP-hard problem, aims to optimize the selection of relevant features without compromising dataset functionality. Hybrid metaheuristic algorithms have shown promise in addressing this challenge.

Progression of Hybrid Metaheuristic Algorithms: The first application of a hybrid metaheuristic algorithm in FS was introduced in 2004 [48], combining Genetic Algorithms (GA) with local search techniques. Since then, various hybrid models have emerged. For instance, [49] proposed a hybrid Whale Optimization Algorithm (WOA) combined with Simulated Annealing (SA). Similarly, [50] developed a hybrid WOA-flower pollination approach for email spam detection. In [51], Grey Wolf Optimization (GWO) was hybridized with Harris Hawks Optimization (HHO), achieving superior results on UCI datasets.

Other studies explored applications beyond FS, such as structural design optimization using hybrid GWO-WOA [52] and data clustering with multi-objective methods [53]. These advancements highlight the adaptability of hybrid algorithms but expose limitations like computational overhead and scalability issues, particularly with high-dimensional datasets. The study in [54] proposes a multi-objective model selection algorithm for the online sequential ultimate learning machine (OSULM) to enhance data transmission and classification accuracy. The algorithm demonstrates superior accuracy and low error rates in selecting multi-objective models through simulation experiments. A study in [55] introduces a top-k feature selection framework leveraging robust 0-1 integer programming. The approach ensures the selection of the most relevant features by formulating the problem as an optimization task, considering robustness against data perturbations. Authors in [56] present a hybrid framework combining Robust Data Optimization (RDO) and XGBoost to improve feature selection and predictive accuracy in cancer classification. The approach leverages RDO to handle noisy data and XGBoost for robust model training. They also came up with a metaheuristic hybrid combining Cuckoo Search and Harris Hawks Optimization for enhanced feature selection in [57]. By identifying optimal feature subsets, the model achieves superior performance in classifying cancer data. The study by Shi et al. [58] applies an evolutionary machine learning approach, integrating a joint self-adaptive sime mould algorithm, to predict recurrent spontaneous abortion (RSA). It leverages medical and biological data to improve diagnostic accuracy and decision-making in RSA cases.

The above-stated literature depicts the success of these hybrid approaches. Nonetheless, the "No Free Lunch" (NFL) theorem in optimization [59] proves logically that, with so many problems out in the research domain, no single optimization algorithm can solve all such problems. Relating it to our context of FS, no algorithm can find the optimal set of features for every dataset. Because of this, attempted to hybridize and improve two of the most recent and popular algorithms, GWO and WOA, to solve binary FS problems over textual data more effectively. As a final summary, existing metaheuristic-based methods suffer from high computational complexity and scalability issues, particularly when

Table 1 Dataset description

Dataset	Low	Medium	High	Critical
MF	164	897	296	516
MT	22	372	164	363
MS	13	291	61	333
MF ESR	10	284	175	203
MT ESR	1	69	15	131

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2021-38502			Exec Code	2021-11-03	2022-07-12	4.3	None	Remote	Medium	Not required	Partial	None	None

Thunderbird ignored the configuration to require STARTTLS security for an SMTP connection. A MITM could perform a downgrade attack to intercept transmitted messages, or could take control of the authenticated session to execute SMTP commands chosen by the MITM. If an unprotected authentication method was configured, the MITM could obtain the authentication credentials, too. This vulnerability affects Thunderbird < 91.2.

Fig. 1 Example of vulnerability data

applied to high-dimensional datasets or large-scale real-world problems. Secondly, most existing work focuses on single-stage hybridization, which may not fully exploit the complementary strengths of different algorithms. Additionally, the parallelization of metaheuristics, which can significantly enhance performance in large-scale FS tasks, is still underexplored, particularly in the context of hybrid methods. The gap this paper aims to address lies in these limitations. The proposed multi-stage framework introduces a novel multi-stage hybridization of metaheuristics to optimize FS performance while leveraging parallelization to enhance scalability and reduce computational costs.

5 Preliminaries

In this section, the basic concepts used to develop the SVSPM are explained.

5.1 Data collection and data preprocessing

This study uses vulnerability reports from five different Mozilla products as datasets. These products include Mozilla Firefox (MF), Mozilla Seamonkey (MS), Mozilla Thunderbird (MT), Mozilla Firefox ESR (MF ESR), and Mozilla Thunderbird ESR (MT ESR). The vulnerability reports are sourced from the CVE database [11], which adheres to the CVSS framework [10] to assign severity scores to vulnerabilities. These scores range from 0 to 10. According to the most recent version of CVSS (v3.0), vulnerabilities with a score of 0.0 receive a 'None' rating, while scores between 0.1 and 3.9 are rated as 'Low.' Vulnerabilities scoring between 4 and 6.9 are rated as 'Medium,' those with scores from 7 to 8.9 are rated as 'High,' and scores from 9 to 10 are classified as 'Critical.' Table 1 below gives the dataset's details in context with severity labels. Out of all the attributes of a vulnerability report, vulnerability description is used to carry out experiments in this study as it is unstructured data. Mining unstructured data

is a challenge as it is vague and may contain noise. Figure 1 gives an example of a vulnerability and all the attributes present in the vulnerability report related to that vulnerability.

Before working with the data, it needs to be pre-processed so that it becomes suitable to be given as input to the classifier using ML algorithms. In our case, we have unstructured data. This unstructured data is pre-processed using standard techniques: tokenization, stop word removal, and stemming. In tokenization, a sequence of words is divided into individual, meaningful words termed tokens. In stop word removal, various words that do not hold any important information and act as connectors helping with framing the sentences are removed from the text [60]. Stemming helps in reducing any word to its root form, thus reducing duplication [61].

5.2 Feature selection approaches

5.2.1 Grey wolf optimization

Grey Wolf Optimization (GWO) is a newly introduced evolutionary algorithm. It is proposed by Mirjalili et al. [22]. This algorithm's foundation is based on the social structure and hunting habits of grey wolves. Wolfs are classified into four hierarchy levels in the GWO algorithm: alpha, beta, delta, and omega. The alpha group represents the ideal answer, the beta group the ideal second-best option, and the delta group the ideal third-best solution. The remaining options fall within the category of omega. Encircling the prey, hunting the prey, and then attacking the prey are the three stages of the GWO approach. The following equation can be used to quantitatively represent the initial stage of encircling the prey:

$$\vec{Y}(i+1) = \vec{Y}_t(i) + \vec{P}.\vec{R} \tag{1}$$

$$\vec{R} = |\vec{Q}.\vec{Y}_t(i) - \vec{Y}(\mathrm{i})| \tag{2}$$

where the wolf vector and the prey vector's positions at iteration i are denoted by $\vec{Y}$ and $\vec{Y}_t$ respectively.

$$\vec{P} = 2\vec{p}.\vec{n}_1 - \vec{p} \tag{3}$$

$$\vec{Q} = 2.\vec{n_2} \tag{4}$$

$\vec{n}_1$ and $\vec{n}_2$ are vectors are vectors that were produced at random in the range [0, 1] and $\vec{p}$ decreases linearly from 2 to 0 over iterations.

In the second stage i.e., hunting, alpha denotes the most optimal solution, idea about the possible position of prey is given by beta and delta. These three best solutions encourage omega to modify their positions in accordance with the optimal position now accessible in the decision space. This hunting procedure is expressed as follows:

$$\vec{Y}(i+1) = \frac{y_1 + y_2 + y_3}{3} \tag{5}$$

where $\overrightarrow{y_1}$, $\overrightarrow{y_2}$, $\overrightarrow{y_3}$ are the top three options after iteration i. These are calculated as follows:

$$\overrightarrow{y_1} = \overrightarrow{Y_\alpha} - P_1.\left(\overrightarrow{R_\alpha}\right) \tag{6}$$

$$\overrightarrow{y_2} = \overrightarrow{Y_\alpha} - P_1.\left(\overrightarrow{R_\alpha}\right) \tag{7}$$

$$\overrightarrow{y_3} = \overrightarrow{Y_\delta} - P_3.\left(\overrightarrow{R_\beta}\right) \tag{8}$$

and $\overrightarrow{R_\alpha}$, $\overrightarrow{R_\beta}$ and $\overrightarrow{R_\delta}$ are calculated as follows:

$$\overrightarrow{R_\alpha} = |\overrightarrow{Q_1}.\overrightarrow{P_\alpha}. - \vec{P}| \tag{9}$$

$$\overrightarrow{R_\beta} = |\overrightarrow{Q_2}.\overrightarrow{P_\beta}. - \vec{P}| \tag{10}$$

$$\overrightarrow{R_\delta} = |\overrightarrow{Q_3}.\overrightarrow{P_\delta}. - \vec{P}| \tag{11}$$

The attacking mechanism of a grey wolf is expressed on the basis of vector $\vec{p}$. The range of this vector is [−p, p]. Over iterations, the value of p is lowered linearly from 2 to 0 and is denoted by:

$$\vec{p} = 2 - i.\frac{2}{MIter} \tag{12}$$

where i is the iteration counter and MIter is the maximum number of iterations.

5.2.2 Whale optimization algorithm

Whale Optimization Algorithm (WOA) is also one of the recent algorithms developed in the class of nature-inspired metaheuristic algorithms [23]. This algorithm imitates humpback whales' hunting techniques. The three steps involved are encircling the prey, bubble-net feeding, and searching the prey. In the first step of encircling, whales update their position according and encircle the prey in accordance with the position of the prey. Initially, WOA presumes that the best solution is given by the prey's position. The best search agent is taken as a reference to update the position of the new search agent according to the following equation:

$$\vec{X}(t+1) = \vec{X}^*(t) - \overrightarrow{AD} \tag{13}$$

$$\vec{D} = \left|\vec{C}\vec{X}^{*}(t) - \vec{X}(t)\right| \tag{14}$$

where $\vec{X}$ is the solution for i^{th} iteration, $\vec{X}^{*}$ is the best solution, $\vec{A}$ and $\vec{C}$ are the coefficient vectors which are computed as follows:

$$\vec{A} = 2\vec{a}\vec{r} - \vec{a} \tag{15}$$

$$C_i = 2\vec{r} \tag{16}$$

where $\vec{r}$ is a random vector in [0,1], a parameter is reduced linearly in the range from 2 to 0 according to:

$$\vec{a} = 2\left(1 - \frac{t}{MIt}\right) \tag{17}$$

where t is the number of iterations and MIt is the maximum number of iterations permitted for the optimization.

In the second step of bubble-net feeding, the shrinking encircling strategy as well as the spiral updating position strategy are defined. The accomplishment of the shrinking encircling strategy is achieved by the decrement of $\vec{a}$ in Eq. (17), which reduces the fluctuation range of $\vec{A}$ indirectly. In the spiral updating position strategy, the position of the whale is updated in a spiral flight direction according to the following equation:

$$\vec{X}(t+1) = \vec{D\prime}.e^{bl}.coscos(2\pi l) + \vec{X}^{*}(t) \tag{18}$$

$$\vec{D\prime} = |\vec{X}^{*}(t) - \vec{X}(t)| \tag{19}$$

l is a random number between -1 and 1, b is a constant that determines the shape of the logarithmic spiral, and. is an element-by-element multiplication. According to the following equations, humpback whales in WOA perform these two methods simultaneously with a probability of 50%:

$$\vec{X}(t+1) = \left\{ \begin{array}{c} \vec{X}^{*}(t) - \vec{A}.\vec{D}p < 0.5 \\ \vec{D\prime}.e^{bl}.coscos(2\pi l) + \vec{X}^{*}(t)p \geq 0.5 \end{array} \right\} \tag{20}$$

where [0, 1] is a range and p is a random vector. The site of the search agent is randomly selected in the third stage in order to search for the prey. The search agent moves away from a randomly chosen humpback whale based on parameter A. The mathematical model is given by:

$$\vec{X}(t+1) = \vec{X}^{R} - \vec{A}.\vec{D} \tag{21}$$

$$\vec{D} = |\vec{C}.\vec{X}^{R} - \vec{X}| \tag{22}$$

where $\vec{X}^{R}$ denotes the random whale's position.

To deal with binary optimization problems such as the FS problem [62], the continuous search space is converted to binary using the sigmoid transfer function using Eq. (23).

$$x_{sig} = \frac{1}{1 + e^{-x_i}}, x_{binary} = \{0 if rand < x_{sig} 1 otherwise \tag{23}$$

where x_{sig} is a continuous feature value that is inputted to the sigmoid function, rand is a random number generated from the uniform distribution that belongs to [0,1], and x_{binary}, is the updated binary position. As demonstrated by Abdel-Basset et al. [63] S-shaped transfer functions have been effectively used in optimization algorithms like the Grey Wolf Optimizer.

5.3 Opposition-based learning

The Opposition Based Learning (OBL) approach is first given by Tizhoosh [64]. When literature is searched, it is found that the OBL approach helps in the enhancement of the convergence speed of metaheuristic algorithms [65, 66]. The metaheuristic algorithm finds an initial solution randomly. It further works on the optimization of this initial solution such that the difference between the initial and optimum solution is minimized. As the initial population is chosen randomly over which the initial solution is found, if the random guess is opposite of the optimal solution, the optimization process will take more time to converge, whereas if the random guess is near to optimal solution, the optimization process will finish early as convergence speed would be fast. But in reality, the optimal solution is unknown. Hence generating an opposite solution to the initial solution using the OBL approach, would improve the chances of the algorithm reaching to optimal solution quickly, increasing the convergence speed.

According to this approach, if x is a real number over interval x ∈ [lb, ub], where lb be the lower bound and ub be the upper bound, ~x is the opposite of x which can be generated using the following equation.

$$\sim x = lb + ub - x \tag{24}$$

For multidimensional feature selection problem, where lb = 0 and ub = 1, the Eq. (24) can be rewritten as

$$\sim x_d = lb_d + ub_d - x_d \tag{25}$$

Here subscript d represents that the problem is multidimensional.

5.4 Data balancing strategy

Most of the ML algorithms are designed on the basis that the data over which they will be working would be balanced. If these algorithms are applied to imbalanced data, the results would be skewed, showing biased results towards the class having more instances [67]. We can see from Table 1, that the data we are dealing with is imbalanced. In this study we used SMOTE. SMOTE stands for Synthetic Minority

Oversampling Technique. More details regarding this algorithm can be found in [68].

5.5 Classification algorithms

To carry out the prediction of the experiment conducted in this study, we chose to work with Extreme Gradient Boosting (XGB), Random Forest (RF), Naïve Bayes (NB), and K- Nearest Neighbour (KNN). XGB and RF are chosen as they come under the category of ensemble learners. Ensemble learners combine multiple ML algorithms to achieve good results. The main objective here is to compensate for the error caused due to one ML algorithm with the help of another ML algorithm and get a better-performing model overall [69]. NB and KNN both are nonparametric ML algorithms. This means these algorithms do not need the data to follow any parametric form for the distribution. Due to this flexibility of the nonparametric model, these algorithms generally lead to a good classifier [70].

5.6 Performance measures and statistical tests

In this study, we used the Area under the ROC Curve (AUC) as the performance measure to gauge our experiment conducted. AUC is a widely used performance metric among researchers, as it effectively balances both specificity and sensitivity. It is also considered an ideal choice for handling imbalanced data problems [71]. In the AUC graph, sensitivity values are plotted on the x-axis, and specificity values are plotted on the y-axis. The AUC score ranges from 0 to 1 [72], where a value of 1 indicates perfect model performance. An AUC between 0.9 and 0.99 signifies excellent performance, between 0.8 and 0.89 indicates good performance, between 0.7 and 0.79 reflects fair performance, and an AUC below 0.7 suggests poor model performance [35]. Without proper statistical validation, conclusions drawn from raw performance metrics may be misleading [73]. To address this, appropriate statistical hypothesis tests were applied to evaluate the significance of the observed results. The Friedman test, a non-parametric statistical test, was used to detect differences in performance across multiple related groups or treatments. This test does not assume normality, making it suitable for datasets with unknown or non-normal distributions. In this study, it was applied to compare the performance of multiple models or configurations over repeated experimental runs, ensuring that differences in AUC scores were statistically significant and not due to random variation. Additionally, the Wilcoxon Signed-Rank test, another non-parametric test, was used for paired data. This test is commonly employed to compare two related samples, such as the performance of two models tested on the same dataset. In our study, it served as a post-hoc analysis following the Friedman test, allowing pairwise comparisons to identify which models or treatments significantly outperformed others.

These tests help reduce the chance of drawing wrong conclusions, showing that the differences observed are meaningful.

Table 2 List of parameters used

Repetition of runs	20
Number of iterations	70
Search agents	8
Dimension	Number of features
Search domain	[0 1]
α	0.99
B	0.01

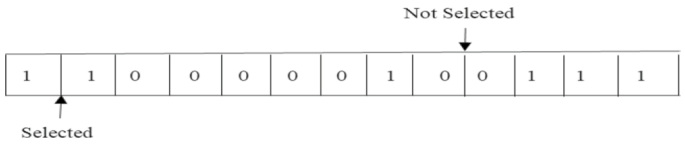

Fig. 2 Solution representation

6 Proposed methodology

The methodology involves the hybridization of GWO with the OBL approach and WOA respectively to generate a hybrid module for FS. This led to the development of two kinds of hybrid modules namely P-GWO and MS-G2WO. The working of the SVSPM developed can be summarized in the following phases, initialization phase, evaluation phase, and feature selection phase. The following subsections will give a thorough explanation of each of these phases.

6.1 Initialization phase

In this study, the input data consists of vulnerability descriptions, which are textual in nature. To analyze this data, it is essential to convert it into vector form using various Natural Language Processing (NLP) techniques. Tokenization breaks the text into smaller units like words or sentences, which is crucial for structuring raw data [74]. Stop word removal eliminates common words like "the" or "is," reducing noise and focusing on relevant terms [75]. Stemming reduces words to their root forms (e.g., "running" becomes "run"), enabling consistent analysis across word variations [76]. Text can then be represented using models like Bag of Words (BoW), which counts word occurrences while ignoring grammar and word order [77]. Finally, TF-IDF (Term Frequency-Inverse Document Frequency) weighs terms based on their importance in the document corpus, highlighting distinctive terms and minimizing the influence of common words [78]. These NLP techniques effectively convert textual data into vectors for further analysis and classification, as demonstrated in various studies.

Every algorithm starts with the setting of parameters and initialization of the population, randomly. Table 2 outlines the parameter settings used to conduct this study. All the parameters are either chosen based on domain-specific information, as in the case of the parameters α, β, or by doing small trial simulations, or they are chosen based on what is commonly found in the literature, as in the case of the remaining parameters. Our primary goal is to compare the performance of the various feature selection problems compared to the proposed module for feature selection.

FS selects the features that boost and maximize the performance of the classifier. Binary chromosome illustration is utilized to depict the feature subset selection as depicted in Fig. 2. The chromosome's length indicates the total number of features. The value '1' suggests selecting the feature and the value '0' suggests rejecting the feature.

6.2 Evaluation phase

FS is considered to be a multi-objective optimization problem focusing on 1) minimizing the count of features, and 2) maximizing the classification accuracy of the SVSPM. Therefore, there is a need for a classification algorithm for the formulation of the fitness functions. K- Nearest Neighbour (K-NN = 5 [79]) is used because of its simplicity, easy implementation, and low computation cost [80–82]. Hence, we can say that a wrapper approach is used for FS making use of the K-NN classifier. The fitness function used to evaluate the solution for FS is depicted in Eq. (26)

$$Fitness(X) = (\alpha * ErrorRate) + (1 - \alpha)\frac{N - S}{S} \tag{26}$$

where X is the binary feature vector, ErrorRate is the classification error rate of the K-NN classifier on the training dataset N is the total number of features, and S is the size of the features subset. α is the balance factor between the selected feature size and the error rate. The choice of $\alpha = 0.99$ prioritizes minimizing classification error, making accuracy the main focus. The smaller $\beta = 1 - \alpha = 0.01$ ensures that feature count is considered but not prioritized, allowing larger subsets if they improve performance. This balance avoids overfitting while maintaining high accuracy, striking a good trade-off between feature count and classifier performance. If the evaluation function focuses solely on classification accuracy, it will overlook solutions that achieve the same accuracy but with fewer selected features, which are crucial for addressing the dimensionality problem. The value of the fitness function is minimized until the termination condition is false.

In this study, the partitioning of the dataset into training and testing sets is done based on a tenfold cross-validation method. The training set is used to train the classifier and the test set is used to check the performance of the classifier [83].

6.3 Feature selection optimization using P-GWO

Having no internal control parameter, GWO is an optimization method that is simple to use. Moreover, GWO has a rational balance between exploration and exploitation through the setting of the parameter 'a' which controls the exploration/exploitation rates at each iteration.

However, the generation of the inappropriate primary population can cause the algorithm to converge less toward the optimal solution. GWO algorithm has a tendency that the local solution will become stagnant [22], therefore its exploration needs to be improved. The approach based on Opposition Based Learning (OBL) is used to initialize the initial population. The main principle of OBL is to find the opposite solution of the current solution and select the best features according to the value of the fitness function. This approach leads to a population rich in diversity giving a good groundwork for the improvement of convergence speed hence aiming to find the best solution fast. As illustrated in Fig. 4a, once the initial solution is found by GWO, the OBL approach is used and the opposite solution is found using Eq. (25). Finally form the set of initial and opposite solutions, the best solution is retained based on its fitness value. For example, If the initial solution had p features and the OBL-based opposite solution had q features, then out of p + q hybrid features, then p is selected as the final solution based on the fitness function. Lastly, the transformation function as specified in Eq. (23) is applied to convert the continuous values to binary. To justify that this approach led to a population rich in diversity, population diversity analysis is done by measuring the moment of inertia which is calculated using Eq. (27)

$$I_c = \sum_{j=1}^{d} \sum_{i=1}^{N} (X_{ji} - c_j)^2 \tag{27}$$

where I_c represents the dispersion of each individual relative to their centroid c_j, d is the dimension of the problem, N is the population size, X_{ji} denotes the jth dimension

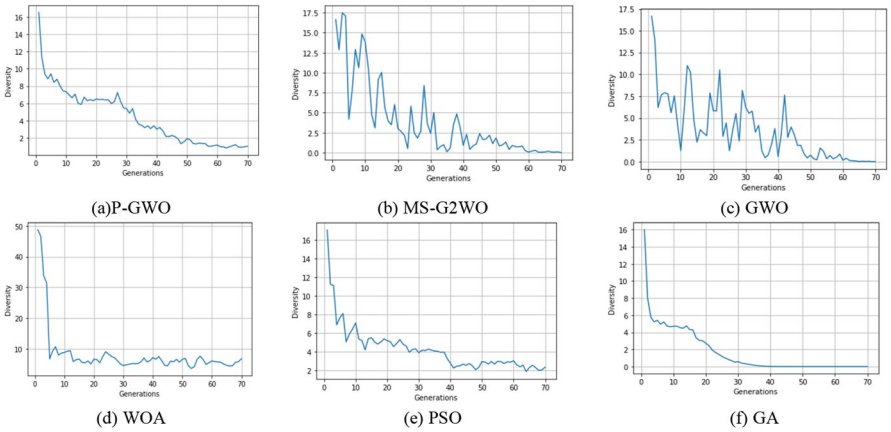

Fig. 3 Population diversity graph

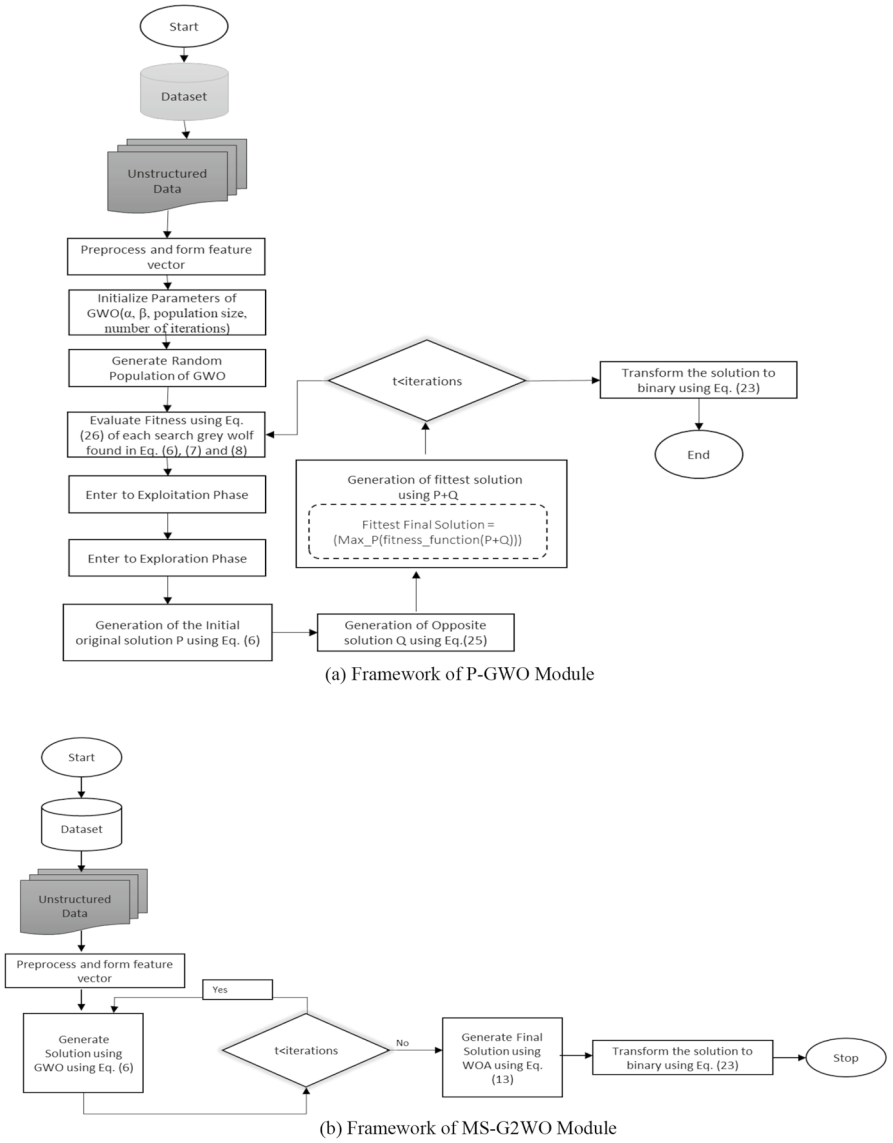

Fig. 4 **a** Framework of P-GWO Module. **b** Framework of MS-G2WO Module

of the ith solution and c_j is the centroid which is calculated using Eq. (28). Figure 3 depicts the population diversity graph.

$$c_j = \frac{1}{N} \sum_{I=1}^{N} X_{JI} \tag{28}$$

It can be interpreted from the graph of Fig. 3 that P-GWO, MS-G2WO, and GWO show the diversity of a population over several generations, but their trends are different. P-GWO demonstrates a more gradual convergence, where diversity decreases over time but does not immediately vanish. This can indicate a healthy exploration of the search space before convergence.MS-G2WO and GWO show a more erratic reduction in diversity, with a tendency to explore more drastically in the early stages but converge quickly to low diversity. WOA maintains moderate diversity after an initial sharp drop, suggesting sustained exploration, which may help avoid premature convergence but could slow down finding an optimal solution. PSO and GA show rapid declines in diversity, with PSO indicating early convergence, while GA shows a slower but more controlled reduction. However, both seem to reach near-zero diversity, which could hinder exploration eventually.

6.4 Feature selection optimization using MS-G2WO

In this sub-section, the proposed MS-G2WO module is elaborated for handling the FS issue. Our hybrid MS-G2WO module uses the WOA algorithm after applying the GWO algorithm and finding the optimum solution. 40 iterations of GWO, followed by 30 iterations of WOA were executed. This approach ensures alignment across all methods while fully utilizing the advantages of sequential hybridization within MS-G2WO. As can be seen in Fig. 4b, WOA is used to enhance the final solution found by GWO. Lastly, the transformation function as specified in Eq. (23) is applied to convert the continuous values to binary.

Results are recorded for the SVSPMs which are developed using the proposed module for FS. To prove that the SVSPM developed using the proposed FS modules is more efficient, results are recorded for other SVSPMs that are developed using PSO, GA, GWO, and WOA for FS. Once the results are recorded for all the SVSPMs, result analysis is performed. For hypothesis testing, statistical analysis is performed to judge whether the performance of different strategies differs significantly or not. For this purpose, we need to frame a null hypothesis (Ho) and an alternate hypothesis (H1).

Friedman test and Wilcoxon Signed Rank Test are applied according to the characteristics of data. Details regarding the application of these tests are documented in the next section of the paper. The result analysis will confirm the efficiency of the proposed modules for FS.

7 Experimental result and analysis

After experimentation, results are recorded. According to these results, in this section, we have addressed the RQs which are formulated and stated above in the introduction section of this paper.

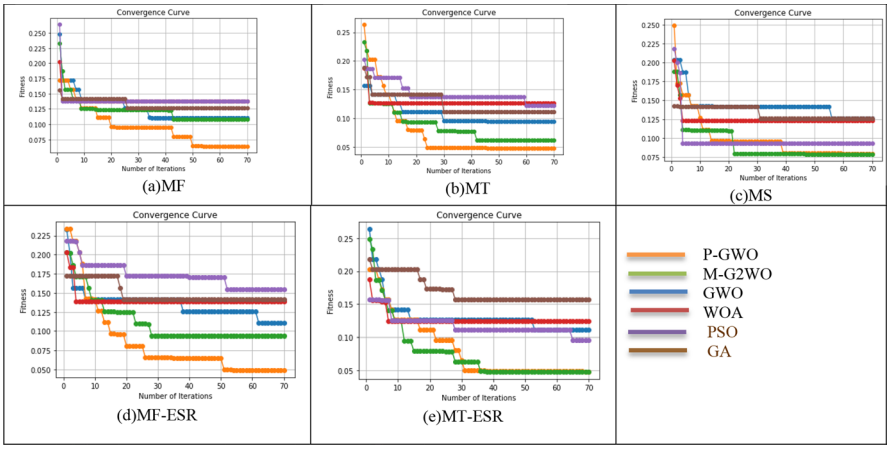

Fig. 5 Convergence Curve of GA, PSO, GWO, WOA, MS-G2WO, P-GWO on Dataset **a** MF **b** MT **c** MS **d** MFESR **d**MT-ESR

7.1 RQ1. What is the improvement in the performance of SVSPM when proposed FS modules are applied to reduce the dimension of the data?

This RQ is divided into three sub-RQs RQ1.1, RQ1.2, and RQ1.3. These RQs will help us to analyze the effectiveness of our proposed module by comparing its performance when no FS approach is used, when native algorithms are used as well as when state-of-the-art algorithms are used, proving that our proposed module gave better performance compared to all. The convergence characteristics of all 6 algorithms over 5 datasets are shown in Fig. 5. From Fig. 5, we can observe that P-GWO converges to minimum fitness function value, giving optimal solutions to the FS problem.

RQ1.1. What is the improvement in the performance of SVSPM when the results of proposed modules P-GWO and MS-G2WO are compared when all features are used and no FS is done?

From Tables 3, 4, 5, and 6 we can see the AUC values which are recorded for the models developed using the proposed modules for FS and the model developed without using FS. Looking at these values it is evident that the SVSPM developed using the proposed modules for FS produced much better results when compared to the SVSPM developed without any FS. The highest AUC values, severity level-wise are made bold. Using these AUC values, columns second and third of Tables 7, 8, 9, and 10 are constructed, having percentage improvement values recorded which are found using the mean AUC values from Tables 3, 4, 5, and 6.

Looking at these values, it is found that for MF and MS, 25.68% and 34.35% are the highest percentage improvement values recorded respectively using the combination of the NB classifier and P-GWO module for FS. Similarly, for MT and MF ESR, 23.13% and 29.62% are the highest percentage improvement values recorded respectively using the combination of the XGB classifier and P-GWO module for FS. For MT ESR, 40.07% is the highest percentage improvement value recorded

Table 3 AUC values obtained from No FS, PSO, GA, GWO, WOA, P-GWO, and MS-G2WO modules for XGB classifier

Dataset	FS approach	Low	Medium	High	Critical	Mean
MF	No FS	0.87	0.63	0.76	0.82	0.77
	PSO	0.89	0.86	0.80	0.92	0.87
	GA	0.87	0.87	0.82	0.90	0.87
	GWO	0.87	0.84	0.75	0.94	0.85
	WOA	0.85	0.88	0.81	0.94	0.87
	P-GWO	0.90	**0.93**	**0.92**	**0.95**	**0.93**
	MS-G2WO	**0.92**	0.86	0.81	0.93	0.88
MT	No FS	0.73	0.61	0.76	0.84	0.74
	PSO	0.81	0.86	0.84	0.90	0.85
	GA	0.75	0.88	**0.87**	0.90	0.85
	GWO	0.77	0.90	0.83	0.94	0.86
	WOA	0.82	0.86	0.86	0.95	0.87
	P-GWO	**0.92**	0.91	0.83	**0.96**	**0.91**
	MS-G2WO	0.81	**0.92**	**0.87**	0.95	0.89
MS	No FS	0.71	0.66	0.62	0.63	0.66
	PSO	0.73	0.91	0.71	0.90	0.81
	GA	0.81	0.89	0.60	0.94	0.81
	GWO	0.80	0.92	0.62	0.90	0.81
	WOA	0.74	0.85	0.58	0.88	0.76
	P-GWO	0.88	**0.92**	**0.78**	**0.94**	**0.88**
	MS-G2WO	**0.89**	0.88	0.71	0.93	0.85
MF ESR	No FS	0.52	0.70	0.77	0.88	0.72
	PSO	0.67	0.81	0.81	0.93	0.81
	GA	0.94	0.86	0.85	0.94	0.90
	GWO	0.90	0.87	0.82	0.93	0.88
	WOA	0.90	0.85	0.85	0.88	0.87
	P-GWO	0.95	**0.88**	**0.92**	**0.97**	**0.93**
	MS-G2WO	**0.98**	**0.88**	0.85	0.93	0.91
MT ESR	No FS	0.49	0.67	0.84	0.72	0.68
	PSO	0.88	0.85	0.88	0.91	0.88
	GA	0.88	0.91	0.93	0.94	0.92
	GWO	0.87	0.84	0.74	0.91	0.84
	WOA	0.85	0.88	0.77	0.92	0.86
	P-GWO	0.90	0.90	**0.98**	0.95	0.93
	MS-G2WO	**0.95**	**0.94**	0.95	**0.97**	**0.95**

Bold values indicate the highest AUC values, severity level-wise

using the combination of the XGB classifier and MS-G2WO module. Therefore, overall, it can be stated that the G2WO module enhanced the SVSPM's performance over all datasets.

In general, it can be stated that the proposed module for FS obtained such a feature subset, which enhanced the performance of the SVSPM. Therefore, the G2WO

Table 4 AUC values obtained from No FS, PSO, GA, GWO, WOA, P-GWO, and MS-G2WO modules for RF classifier

Dataset	FS approach	Low	Medium	High	Critical	Mean
MF	No FS	0.87	0.63	0.76	0.82	0.77
	PSO	0.87	0.86	0.76	0.91	0.85
	GA	0.91	0.88	0.82	0.95	0.89
	GWO	0.83	0.85	0.77	0.93	0.85
	WOA	0.80	0.89	0.78	0.96	0.86
	P-GWO	0.90	**0.93**	**0.92**	**0.95**	**0.93**
	MS-G2WO	**0.92**	0.86	0.81	0.93	0.88
MT	No FS	0.73	0.61	0.76	0.84	0.74
	PSO	0.72	0.87	0.86	0.93	0.85
	GA	0.72	0.85	0.88	0.91	0.84
	GWO	0.82	0.85	0.88	0.93	0.87
	WOA	0.84	0.86	0.86	0.93	0.87
	P-GWO	**0.92**	0.91	0.83	**0.96**	**0.91**
	MS-G2WO	0.81	**0.92**	**0.87**	0.95	0.89
MS	No FS	0.71	0.66	0.62	0.63	0.66
	PSO	0.83	0.93	0.78	0.93	0.87
	GA	0.73	0.89	0.65	0.90	0.79
	GWO	0.87	0.95	0.76	0.95	0.88
	WOA	0.61	0.82	0.65	0.90	0.75
	P-GWO	0.88	**0.92**	**0.78**	**0.94**	**0.88**
	MS-G2WO	**0.89**	0.88	0.71	0.93	0.85
MF ESR	No FS	0.52	0.70	0.77	0.88	0.72
	PSO	0.78	0.88	0.86	0.91	0.86
	GA	0.91	0.83	0.85	0.94	0.88
	GWO	0.79	0.85	0.83	0.91	0.85
	WOA	0.92	0.83	0.83	0.90	0.87
	P-GWO	0.95	**0.88**	**0.92**	**0.97**	**0.93**
	MS-G2WO	**0.98**	**0.88**	0.85	0.93	0.91
MT ESR	No FS	0.49	0.67	0.84	0.72	0.68
	PSO	0.42	0.91	0.78	0.92	0.76
	GA	0.39	0.93	0.77	0.96	0.76
	GWO	0.47	0.95	0.93	0.98	0.83
	WOA	0.89	0.87	0.94	0.93	0.91
	P-GWO	0.90	0.90	**0.98**	0.95	0.93
	MS-G2WO	**0.95**	**0.94**	0.95	**0.97**	**0.95**

Bold values indicate the highest AUC values, severity level-wise

module can be successfully used for FS reducing the dimensionality and enhancing the SVSPM's performance.

RQ1.2. What is the improvement in the performance of SVSPM when the results of proposed modules P-GWO and MS-G2WO are compared with state-of-the-art algorithms?

Table 5 AUC values obtained from No FS, PSO, GA, GWO, WOA, P-GWO, and MS-G2WO modules for NB classifier

Dataset	FS approach	Low	Medium	High	Critical	Mean
MF	No FS	0.87	0.63	0.76	0.82	0.77
	PSO	0.60	0.61	0.63	0.71	0.64
	GA	0.65	0.66	0.66	0.73	0.68
	GWO	0.64	0.60	0.59	0.73	0.64
	WOA	0.61	0.64	0.61	0.73	0.65
	P-GWO	0.90	**0.93**	**0.92**	**0.95**	**0.93**
	MS-G2WO	**0.92**	0.86	0.81	0.93	0.88
MT	No FS	0.73	0.61	0.76	0.84	0.74
	PSO	0.49	0.67	0.68	0.72	0.64
	GA	0.49	0.66	0.67	0.74	0.64
	GWO	0.61	0.68	0.65	0.73	0.67
	WOA	0.59	0.61	0.62	0.64	0.62
	P-GWO	**0.92**	0.91	0.83	**0.96**	**0.91**
	MS-G2WO	0.81	**0.92**	**0.87**	0.95	0.89
MS	No FS	0.71	0.66	0.62	0.63	0.66
	PSO	0.50	0.65	0.60	0.59	0.59
	GA	0.50	0.67	0.68	0.68	0.63
	GWO	0.47	0.65	0.64	0.64	0.60
	WOA	0.50	0.74	0.65	0.72	0.65
	P-GWO	0.88	**0.92**	**0.78**	**0.94**	**0.88**
	MS-G2WO	**0.89**	0.88	0.71	0.93	0.85
MF ESR	No FS	0.52	0.70	0.77	0.88	0.72
	PSO	0.66	0.66	0.66	0.72	0.68
	GA	0.66	0.70	0.72	0.74	0.71
	GWO	0.66	0.66	0.72	0.70	0.69
	WOA	0.66	0.68	0.72	0.80	0.72
	P-GWO	0.95	**0.88**	**0.92**	**0.97**	**0.93**
	MS-G2WO	**0.98**	**0.88**	0.85	0.93	0.91
MT ESR	No FS	0.49	0.67	0.84	0.72	0.68
	PSO	0.50	0.70	0.67	0.68	0.64
	GA	0.50	0.67	0.79	0.79	0.69
	GWO	0.50	0.72	0.64	0.63	0.62
	WOA	0.85	0.27	0.44	0.77	0.58
	P-GWO	0.90	0.90	**0.98**	0.95	0.93
	MS-G2WO	**0.95**	**0.94**	0.95	**0.97**	**0.95**

Bold values indicate the highest AUC values, severity level-wise

The impact of other popular metaheuristic algorithms such as PSO and GA on FS is analysed and compared with the proposed modules for FS. PSO parameters include acceleration constants c1 and c2 which usually take a value of 2, inertia weight w = 0.9 [84] with population quantity as 20 and maximum number of iterations as 70. In GA, we set a 0.6 crossover rate and a 0.2 mutation rate. Typically,

Table 6 AUC values obtained from No FS, PSO, GA, GWO, WOA, P-GWO and MS-G2WO modules for KNN classifier

Dataset	FS approach	Low	Medium	High	Critical	Mean
MF	No FS	0.87	0.63	0.76	0.82	0.77
	PSO	0.78	0.69	0.72	0.89	0.77
	GA	0.84	0.75	0.72	0.85	0.79
	GWO	0.74	0.73	0.71	0.84	0.76
	WOA	0.70	0.63	0.71	0.89	0.73
	P-GWO	0.90	**0.93**	**0.92**	**0.95**	**0.93**
	MS-G2WO	**0.92**	0.86	0.81	0.93	0.88
MT	No FS	0.73	0.61	0.76	0.84	0.74
	PSO	0.79	0.65	0.85	0.89	0.80
	GA	0.78	0.73	0.80	0.87	0.80
	GWO	0.60	0.81	0.81	0.89	0.78
	WOA	0.74	0.69	0.84	0.85	0.78
	P-GWO	**0.92**	0.91	0.83	**0.96**	**0.91**
	MS-G2WO	0.81	**0.92**	**0.87**	0.95	0.89
MS	No FS	0.71	0.66	0.62	0.63	0.66
	PSO	0.65	0.81	0.74	0.88	0.77
	GA	0.52	0.71	0.66	0.84	0.68
	GWO	0.56	0.83	0.74	0.88	0.75
	WOA	0.57	0.87	0.59	0.89	0.73
	P-GWO	0.88	**0.92**	**0.78**	**0.94**	**0.88**
	MS-G2WO	**0.89**	0.88	0.71	0.93	0.85
MF ESR	No FS	0.52	0.70	0.77	0.88	0.72
	PSO	0.98	0.72	0.83	0.91	0.86
	GA	0.98	0.82	0.85	0.89	0.89
	GWO	0.98	0.84	0.87	0.90	0.90
	WOA	0.97	0.81	0.89	0.92	0.90
	P-GWO	0.95	**0.88**	**0.92**	**0.97**	**0.93**
	MS-G2WO	**0.98**	**0.88**	0.85	0.93	0.91
MT ESR	No FS	0.49	0.67	0.84	0.72	0.68
	PSO	0.50	0.83	0.76	0.86	0.74
	GA	0.65	0.90	0.88	0.94	0.84
	GWO	0.49	0.90	0.43	0.94	0.69
	WOA	0.46	0.88	0.88	0.95	0.79
	P-GWO	0.90	0.90	**0.98**	0.95	0.93
	MS-G2WO	**0.95**	**0.94**	0.95	**0.97**	**0.95**

Bold values indicate the highest AUC values, severity level-wise

crossover rates between 0.6 and 0.9 work well, as higher rates encourage exploration but may cause premature convergence if too high [85]. A mutation rate of 0.01–0.05 is usually preferred, but for feature selection, slightly higher values (0.1–0.2) help explore more feature combinations efficiently [86]. It is clear from the AUC values recorded in Tables 3, 4, 5, and 6 that the proposed module outperformed PSO

Table 7 Percentage improvement of proposed modules Vs. other FS approaches for XGB classifier

Dataset	No FS vs P-GWO	No FS vs MS-G2WO	PSO vs P-GWO	PSO vs MS-G2WO	GA vs P-GWO	GA vs MS-G2WO	GWO vs P-GWO	GWO vs MS-G2WO	WOA vs P-GWO	WOA vs MS-G2WO
MF	**20.13**	14.29	**6.63**	1.44	**6.94**	1.73	**8.91**	3.62	**6.32**	1.15
MT	**23.13**	20.75	**6.16**	4.11	**6.47**	4.41	**5.23**	3.20	**3.72**	1.72
MS	**34.35**	30.15	**8.31**	4.92	**8.64**	5.25	**8.64**	5.25	**15.41**	11.80
MF ESR	**29.62**	26.83	**15.53**	13.04	**3.62**	1.39	**5.68**	3.41	**6.90**	4.60
MT ESR	37.13	**40.07**	5.97	**8.24**	1.91	**4.10**	11.01	**13.39**	9.06	**11.40**

The percentage improvement of the 2 proposed modules compared to other FS approachesis recorded. For each table, the proposed module out of the 2 that achieves the higher percentage improvement hasits value highlighted in bold

Table 8 Percentage improvement of proposed modules Vs. other FS approaches for RF classifier

Dataset	No FS vs P-GWO	No FS vs MS-G2WO	PSO vs P-GWO	PSO vs MS-G2WO	GA vs P-GWO	GA vs MS-G2WO	GWO vs P-GWO	GWO vs MS-G2WO	WOA vs P-GWO	WOA vs MS-G2WO
MF	**4.66**	4.08	**5.59**	5.00	**0.84**	0.28	**6.21**	5.62	**4.66**	4.08
MT	**9.20**	7.12	**8.88**	6.80	**9.52**	7.44	**5.75**	3.74	**5.44**	3.44
MS	**−2.86**	−3.43	**−2.02**	−2.59	**7.26**	6.62	**−3.68**	−4.25	**14.09**	13.42
MF ESR	1.38	**1.65**	7.29	**7.58**	4.25	**4.53**	8.88	**9.17**	5.75	**6.03**
MT ESR	16.09	**22.08**	21.45	**27.72**	20.66	**26.89**	10.51	**16.22**	1.38	**6.61**

The percentage improvement of the 2 proposed modules compared to other FS approachesis recorded. For each table, the proposed module out of the 2 that achieves the higher percentage improvement hasits value highlighted in bold

Table 9 Percentage improvement of proposed modules Vs. other FS approaches for NB Classifier

Dataset	No FS vs P-GWO	No FS vs MS-G2WO	PSO vs P-GWO	PSO vs MS-G2WO	GA vs P-GWO	GA vs MS-G2WO	GWO vs P-GWO	GWO vs MS-G2WO	WOA vs P-GWO	WOA vs MS-G2WO
MF	**25.68**	19.07	**26.67**	20.00	**19.63**	13.33	**26.17**	19.53	**24.71**	18.15
MT	**22.81**	18.25	**26.17**	21.48	**26.17**	21.48	**20.97**	16.48	**31.30**	26.42
MS	**35.22**	26.09	**32.91**	23.93	**22.92**	14.62	**29.58**	20.83	**19.16**	11.11
MF ESR	**12.10**	10.32	**16.67**	14.81	**11.70**	9.93	**14.96**	13.14	**10.14**	8.39
MT ESR	**29.62**	22.31	**32.16**	24.71	**22.55**	15.64	**35.34**	27.71	**44.64**	36.48

The percentage improvement of the 2 proposed modules compared to other FS approachesis recorded. For each table, the proposed module out of the 2 that achieves the higher percentage improvement hasits value highlighted in bold

Table 10 Percentage improvement of proposed modules Vs. other FS approaches for KNN Classifier

Dataset	No FS vs P-GWO	No FS vs MS-G2WO	PSO vs P-GWO	PSO vs MS-G2WO	GA vs P-GWO	GA vs MS-G2WO	GWO vs P-GWO	GWO vs MS-G2WO	WOA vs P-GWO	WOA vs MS-G2WO
MF	**8.50**	6.86	**7.79**	6.17	**5.06**	3.48	**9.93**	8.28	**13.31**	11.60
MT	**28.62**	27.90	**11.64**	11.01	**11.64**	11.01	**14.15**	13.50	**13.78**	13.14
MS	**26.92**	25.77	**7.14**	6.17	**20.88**	19.78	**9.63**	8.64	**13.01**	11.99
MF ESR	**19.23**	19.23	**8.14**	8.14	**5.08**	5.08	**3.62**	3.62	**3.62**	3.62
MT ESR	33.58	**39.05**	24.07	**29.15**	8.61	**13.06**	32.61	**38.04**	15.46	**20.19**

The percentage improvement of the 2 proposed modules compared to other FS approachesis recorded. For each table, the proposed module out of the 2 that achieves the higher percentage improvement hasits value highlighted in bold

as well as GA, producing better SVSPM having higher AUC values. Percentage improvement is recorded in the fourth, fifth, sixth, and seventh columns of Tables 7, 8, 9, and 10.

It is found that, from PSO, 26.67%, 26.17%, 32.91%, 16.67%, and 32.16% are the highest percentage improvement values for all five datasets respectively. These values came when the combination of the NB classifier and P-GWO FS module is used. Therefore, this gives us an overall view that the proposed module gave better-performing SVSPMs compared to native algorithms. Similarly, from GA, 19.63%, 26.17%, 22.92%, 11.70%, and 22.55% are the highest percentage improvement values for all five datasets respectively. These values came when the combination of the NB classifier and P-GWO FS module is used. Therefore, this gives us an overall view that the proposed module gave better-performing SVSPMs compared to native algorithms.

From the above discussion, it can be stated that the proposed module for FS obtained such a feature subset, which enhanced the performance of the SVSPM. Therefore, the G2WO module can be successfully used for FS. The SVSPM's performance with the G2WO module is improved and the dimensionality is also reduced.

7.2 RQ2. Which module of FS among the two proposed modules gave better results?

In this RQ we tried to gauge and compare the performance of SVSPM developed using the two proposed modules. To do so we looked at the overall percentage improvement values which are found using mean AUC values. Tables 7, 8, 9, and 10 can be referred for that. These four tables contain a total of two hundred values. Out of these two hundred values, a hundred values are of percentage improvement observed from P-GWO and the remaining hundred values are of percentage improvement observed from MS-G2WO. A common pattern is observed when these values are analyzed. This common pattern can be visualized by looking at the values that are made bold in Tables 7, 8, 9, and 10. For SVSPM developed using the XGB classifier, P-GWO gave a higher percentage improvement value for MF, MT, MS, and MF ESR as well as for all methods for FS. MS-G2WO gave a higher percentage improvement value for only MT ESR. For SVSPM developed using an RF classifier, P-GWO gave a higher percentage improvement value for MF, MT, and MS, as well as for all methods for FS. MS-G2WO gave a higher percentage improvement value for MF ESR and MT ESR. For SVSPM developed using the NB classifier, P-GWO gave a higher percentage improvement value for all five datasets as well as for all methods for FS. MS-G2WO gave a higher percentage improvement value only for MT ESR For SVSPM developed using the KNN classifier, P-GWO gave a higher percentage improvement value for MF, MT, MS, and MF ESR as well as for all methods for FS. MS-G2WO gave a higher percentage improvement value for only MT ESR. As the majority of values (eighty) are for P-GWO and only twenty values are for MS-G2WO, it can be concluded that the P-GWO module provides better results.

7.3 RQ3. Whether the performance of SVSPM developed boosted significantly with the help of FS?

To address RQ3, null and alternate hypotheses are formulated as follows:

Ho The performance of SVSPMs developed using different FS approaches is the same and does not differ significantly.

H1 The performance of SVSPMs developed using different FS approaches differs significantly.

Table 11 Friedman test results

Classifier	FS approach	Mean rank (Low)	Mean rank (Medium)	Mean rank (High)	Mean rank (Critical)	Mean rank (Average)
XGB	NO FS	1.50	1.00	2.10	1.10	1.00
	GWO	3.10	4.10	2.20	3.90	3.10
	WOA	3.10	3.50	3.50	3.70	3.60
	PSO	3.60	3.20	3.70	3.10	3.60
	GA	4.00	4.60	4.90	4.40	3.70
	P-GWO	6.20	**6.20**	**6.10**	**6.70**	**6.80**
	MS-G2WO	**6.50**	5.40	5.50	5.10	6.20
RF	NO FS	4.00	5.00	3.60	4.10	3.90
	GWO	3.40	3.70	3.40	5.10	3.40
	WOA	3.30	2.60	3.10	3.30	3.50
	PSO	2.50	4.10	3.10	2.80	2.60
	GA	2.60	2.80	3.30	3.00	2.80
	P-GWO	5.40	**4.90**	**6.70**	4.40	**6.00**
	MS-G2WO	**6.80**	**4.90**	4.80	**5.30**	5.80
NB	NO FS	3.10	2.30	2.50	3.70	3.00
	GWO	2.9	3.3	2.6	2.1	2.8
	WOA	3.6	3.1	2.5	3.8	3.2
	PSO	2.4	3	3	1.6	1.9
	GA	3.2	4	4.6	4	4.1
	P-GWO	**6.6**	**6.9**	**6.6**	**7**	**7**
	MS-G2WO	6.2	5.4	6.2	5.8	6
KNN	NO FS	2.8	1.3	1.8	1.3	1.3
	GWO	2.1	4.7	3	3.1	3.1
	WOA	2.2	3.4	3.4	4.4	4.4
	PSO	4.3	2.3	3.7	3.8	3.8
	GA	4	4.1	3.3	2.6	2.6
	P-GWO	**6.5**	5.6	**6.7**	**6.4**	**6.4**
	MS-G2WO	6.1	**6.6**	6.1	**6.4**	**6.4**

The results of the Friedman test. For each level of severity, the highest mean rank value is highlighted in bold

Table 12 *p*-values obtained using Friedman test

Classifier	Low	Medium	High	Critical	Mean
XGB	0.002	0.005	0.13	0.003	0
RF	0.013	0.317	0.7	0.343	0.6
NB	0.002	0.002	0.002	0.001	0.001
KNN	0.002	0.001	0.002	0	0.001

To test the above-stated hypothesis, α is considered to be 0.05. The meaning of 0.05 is that the confidence level is 95%. Table 11 records the Friedman test's results. The mean rank for all four levels and the average case, for all models are documented databases-wise. A large value of rank obtained by a particular model indicates that that model is better. As we can see, the highest rank is obtained by P-GWO in the majority of the cases (sixteen out of twenty), hence we can say that P-GWO performed best compared to other FS approaches. Thus, this analysis also validates the finding of RQ2. *P*-values are recorded in Table 12. Out of twenty cases, in sixteen cases *p*-value came out to be less than 0.05 indicating that the results of Friedman tests are significant. Therefore, for these sixteen cases, Ho is rejected and H1 is accepted.

As the results of the Friedman tests are found to be significant for sixteen cases, therefore, for those cases, Wilcoxon Signed Rank Test is applied for post hoc analysis. *P*-values for the grey area of Table 13 are not calculated because, in Friedman Test results, the result came out to be not significant. To judge the overall performance, let's concentrate on the *p*-values of the mean case. Looking at the values of Table 13, we can say that SVSPM developed using RF does not produce any significant results. Although SVSPM developed using XGB and NB produced significant results for all pairs. Therefore, it is concluded that the proposed modules for FS outperformed other methods of FS and showed significant improvement in the performance of the model developed.

The proposed P-GWO module enhances computational efficiency by combining Opposition-Based Learning (OBL) with parallel execution, reducing time complexity to O(T/2) and memory usage to O(n), where T is the total iterations and *n* is the population size. This ensures faster convergence and lower resource consumption while maintaining solution quality. On the other hand, the MS-G2WO module combines GWO and WOA in a sequential manner, enhancing solution refinement at the expense of increased memory usage O(2n)**.** Although the time complexity is the same as T, the total number of iterations is the same. Consequently, P-GWO is better suited for environments with strict time and memory constraints.

8 Threat to validity

In this section, the threats to the validity of this empirical study [87] are discussed. Internal validity examines factors that may influence the results of the study. Our experiments were conducted using vulnerability reports from five Mozilla products

Table 13 Pair wise *p*-values obtained using Wilcoxon signed rank test

Classifier	Pair	*P*-val (Low)	*P*-val (Medium)	*P*-val (High)	*P*-val (Critical)	*P*-val (Mean)
XGB	PG2WO-NoFS	**0.043**	**0.042**	**0.042**	**0.043**	**0.043**
	PG2WO-GWO	**0.043**	0.066	0.068	**0.039**	**0.043**
	PG2WO-WOA	**0.039**	**0.042**	0.08	**0.042**	**0.043**
	PG2WO-PSO	**0.043**	**0.041**	0.08	**0.039**	**0.042**
	PG2WO-GA	**0.043**	0.078	0.08	0.068	**0.043**
	MSG2WO-NoFS	**0.042**	**0.043**	**0.042**	**0.042**	**0.043**
	MSG2WO-GWO	**0.042**	0.343	**0.043**	0.197	**0.043**
	MSG2WO-WOA	0.08	0.077	0.109	0.131	**0.043**
	MSG2WO-PSO	0.068	0.144	0.068	0.068	**0.043**
	MSG2WO-GA	**0.043**	0.223	0.285	0.221	**0.042**
RF	PG2WO-NoFS	0.225	–	–	–	–
	PG2WO-GWO	0.225	–	–	–	–
	PG2WO-WOA	0.068	–	–	–	–
	PG2WO-PSO	0.138	–	–	–	–
	PG2WO-GA	0.068	–	–	–	–
	MSG2WO-NoFS	0.043	–	–	–	–
	MSG2WO-GWO	0.043	–	–	–	–
	MSG2WO-WOA	0.043	–	–	–	–
	MSG2WO-PSO	0.043	–	–	–	–
	MSG2WO-GA	0.043	–	–	–	–
NB	PG2WO-NoFS	**0.042**	0.066	**0.043**	0.068	**0.043**
	PG2WO-GWO	**0.043**	**0.043**	**0.042**	**0.043**	**0.043**
	PG2WO-WOA	**0.042**	**0.043**	**0.043**	**0.042**	**0.043**
	PG2WO-PSO	**0.043**	**0.043**	**0.042**	**0.043**	**0.043**
	PG2WO-GA	**0.042**	**0.042**	0.078	**0.043**	**0.043**
	MSG2WO-NoFS	**0.043**	0.08	**0.043**	**0.042**	**0.043**
	MSG2WO-GWO	**0.042**	0.068	**0.042**	**0.043**	**0.042**
	MSG2WO-WOA	0.08	0.068	**0.043**	0.08	**0.043**
	MSG2WO-PSO	**0.043**	0.043	**0.043**	**0.043**	**0.043**
	MSG2WO-GA	**0.042**	0.078	**0.043**	**0.042**	**0.043**

Table 13 (continued)

Classifier	Pair	*P*-val (Low)	*P*-val (Medium)	*P*-val (High)	*P*-val (Critical)	*P*-val (Mean)
KNN	PG2WO-NoFS	0.225	0.414	0.109	0.892	0.138
	PG2WO-GWO	0.225	0.891	**0.042**	0.214	0.08
	PG2WO-WOA	0.068	0.068	**0.043**	0.066	**0.043**
	PG2WO-PSO	0.138	0.496	**0.042**	0.223	0.08
	PG2WO-GA	0.068	0.225	**0.042**	0.141	**0.043**
	MSG2WO-NoFS	**0.043**	1	0.684	0.461	0.138
	MSG2WO-GWO	**0.043**	0.276	0.5	0.705	0.138
	MSG2WO-WOA	**0.043**	0.078	0.498	0.102	**0.043**
	MSG2WO-PSO	**0.043**	0.221	0.5	**0.042**	0.08
	MSG2WO-GA	**0.043**	0.059	0.465	0.074	**0.043**

The pairwise *p*-values obtained using the Wilcoxon signed-rank test. At a 95% confidence level, *p*-values below 0.05 are highlighted in bold

provided by CVE-NVD. A potential threat arises from the possibility of new vulnerabilities being reported that are not included in our collected database, which could impact the validity of the results. Additionally, the CVSS scoring framework was used to assess the severity of software vulnerabilities. However, recently proposed metrics such as the Vulnerability Rating and Scoring System (VRSS) and the Weighted Impact Vulnerability Scoring System (WIVSS) suggest that improvements in severity assessment processes are possible.

External validity considers whether the study's findings can be generalized. In this research, twenty-eight models were developed using seven feature selection (FS) approaches and four classifiers, with their performance evaluated on five Mozilla products. While the results may be generalizable within the domain of Mozilla products, they could vary for other domains. Although several software vulnerabilities scoring prediction models (SVSPMs) exist in the literature, our findings suggest that the models built with the proposed FS module achieved competitive performance. Lastly, the Area Under the Curve (AUC) metric was used to evaluate the effectiveness of the developed SVSPMs. As AUC balances sensitivity and specificity, this reduces threats to the validity of the results.

9 Conclusion and future work

Feature selection (FS) plays a crucial role in enhancing the performance of software vulnerability scoring prediction models (SVSPMs), especially when processing textual data. In this study, we proposed two hybrid FS modules: Parallel-Grey

Wolf Optimization (P-GWO) and Multi-Stage Grey Wolf Whale Optimization (MS-G2WO). The P-GWO module employs a hybridized population generated by the Opposition-Based Learning (OBL) approach and GWO, while MS-G2WO applies GWO and Whale Optimization Algorithm (WOA) sequentially, with WOA refining the solution provided by GWO. These modules aim to improve the quality of the population and, consequently, the final optimum solution.

The performance of the proposed modules was evaluated against other FS approaches, including No-FS, PSO, GA, GWO, and WOA, using the AUC metric, which balances sensitivity and specificity and is particularly suited for imbalanced datasets [58]. Experimental results showed that all FS approaches improved the performance of SVSPMs compared to models without FS. Among them, the P-GWO module demonstrated superior results with an average AUC of 0.804, followed by MS-G2WO with an average AUC of 0.77. The statistical significance of these findings was validated using the Friedman Test, which assigned the highest rank to the P-GWO module.

The findings highlight that FS significantly enhances SVSPM performance by addressing challenges such as high dimensionality and imbalanced data. Beyond the immediate contributions to SVSPMs, these results have broader implications for software engineering practices. By effectively reducing feature dimensionality and improving prediction accuracy, the proposed FS modules can assist practitioners in developing more reliable and scalable vulnerability assessment tools. This can lead to better prioritization of software vulnerabilities, improved resource allocation, and enhanced security practices within the software development lifecycle.

This study opens avenues for refining the proposed methodologies. Promising directions include integrating deep learning techniques like Recurrent Neural Networks (RNNs) and transformers for automatic feature extraction and handling complex relationships in high-dimensional data. Domain-specific applications in NLP tasks, such as sentiment analysis and spam detection, can enhance model efficiency and interpretability.

Future work could also focus on advanced data-balancing techniques and leveraging quantum algorithms like Quantum Approximate Optimization Algorithm (QAOA) for exploring large FS search spaces. Hybrid classical-quantum metaheuristic approaches may provide scalable solutions for NP-hard FS challenges. Additionally, optimizing algorithm parameters could further improve performance.

By exploring these directions, the methodologies can be extended for broader applicability, enabling significant advancements in software vulnerability prediction and software engineering practices.

Author contributions Ruchika Malhotra and Vidushi conceptualized the idea presented in the manuscript. Vidushi carried out the experiment and wrote the first draft of the manuscript. Ruchika Malhotra supervised the work, reviewed the draft and finalized the manuscript. Both authors read and approved the final manuscript after revisions.

Funding The author did not receive support from any organization for the submitted work.

Data availability On reasonable request from the corresponding author of the manuscript, the datasets analyzed in this study will be made available.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. McGraw G (2006) Software security: building security. Addison-Wesley Professional, vol 1
2. Furnell S (2009) Cybercrime in society. Connected Minds, Emerging Cultures: Cybercultures in Online Learning
3. Violettas GE, Theodorou TL, and Stephanides GC (2013) E-learning software security: tested for security vulnerabilities & issues. E-Learning best practices in management, design and development of e-courses: standards of excellence and creativity: 233–240
4. Younan Y (2013) 25 years of vulnerabilities: 1988–2012. Sourcefire vulnerability research team
5. Coulter R, Han QL, Pan L, Zhang J, Xiang Y (2020) Data-driven cyber security in perspective–intelligent traffic analysis. IEEE Transactions on Cybernetics. https://doi.org/10.1109/TCYB.2019.2940940
6. Bilge L, Dumitras T (2012) Before we knew it: an empirical study of zero-day attacks in the real world. ACM Conference on Computer and Communications Security 833–844
7. Microsoft C (2002) Microsoft security response center security bulletin severity rating system. https://technet.microsoft.com/zhcn/security/gg309177.aspx. Accessed 4 October 2022
8. US-CERT (2006) Uscert vulnerability note field descriptions. http://www.kb.cert.org/vuls/html/fieldhelp. Accessed 4 October 2022
9. Sans I (2022) Sans critical vulnerability analysis archive. http://www.sans.org/newsletters/cva/ Accessed 4 October 2022
10. FIRST (2007) Common vulnerability scoring system (cvss) version 2.0. https: //www.first.org/cvss/v2/guide#i1.2 Accessed 4 October 2022
11. Mitre C (2022) Common vulnerabilities and exposures (cve). https://cve.mitre.org/. Accessed 4 October 2022
12. Chizi B, Rokach L, Maimon O (2009) A survey of feature selection techniques. Encyclopedia of data warehousing and mining, second ed. 1888–1895
13. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Elect Eng 40:16–28
14. Dash M, Liu H (1997) Feature selection for classification. Intelligent Data Analysis 1:131–156
15. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
16. Amaldi E, Kann V (1998) On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theoret Comput Sci 209(1–2):237–260
17. Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK, Ryan MJ (2020) An efficient-assembler whale optimization algorithm for DNA fragment assembly problem: analysis and validations. IEEE Access 8:222144–222167
18. Abdel-Basset M, Chang V, Mohamed R (2020) A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. Neural Comput Appl 33:10685–10718
19. Al-Tashi Q, Rais H, and Jadid S (2019) Feature selection method based on grey wolf optimization for coronary artery disease classification. https://doi.org/10.1007/978-3-319-99007-1_25
20. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99
21. Eberhart R, Kennedy J (1995) Particle swarm optimization. IEEE international conference on neural networks 4:1942–1948
22. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
23. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
24. Blum C, Roli A (2008) Hybrid metaheuristics: an introduction. hybrid metaheuristics—an emerging approach to optimization. Studies in Computational Intelligence 114:1–30
25. Mohamed AW, Hafez AI, Abdelaziz AY (2021) A novel hybrid whale and grey wolf optimizer for solving engineering design problems. Soft Comput 25(9):6173–6201
26. Ibrahim RA, Jambek AB, Nor MJ (2022) Hybrid whale-grey wolf optimization for feature selection in medical data analysis. Expert Syst Appl 184:115506

27. Kumar R, Bansal JC, Dhiman G (2020) Hybridization of whale optimization algorithm with grey wolf optimizer for global optimization. J Ambient Intell HumIzed Comput 11(2):911–926
28. Emary E, Zawbaa HM, Hassanien AE (2015) Binary gray wolf optimization approaches for feature selection. Neurocomputing. https://doi.org/10.1016/j.neucom.2015.06.083
29. Al-Tashi Q, Abdulkadir SJ, Rais HM, Mirjalili S, Alhussian H, Ragab MG, Alqushaibi A (2020) Binary multi-objective grey wolf optimizer for feature selection in classification. IEEE Access 8:106247–106263
30. Hu P, Pan JS, Chu SC (2020) Improved binary grey wolf optimizer and its application for feature selection. Knowl-Based Syst 195:105746
31. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453
32. Shuaib M, Abdulhamid SIM, Adebayo OS, Osho O, Idris I, Alhassan JK, Rana N (2019) Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification. SN Applied Sciences 1(5):1–17
33. Got A, Moussaoui A, Zouache D (2021) Hybrid filter-wrapper feature selection using whale optimization algorithm: a multi-objective approach. Expert Syst Appl 183:115312
34. Gupta MK, Govil MC, Singh G (2015) Text-mining based predictive model to detect XSS vulnerable files in web applications. Annual IEEE India Conference. https://doi.org/10.1109/INDICON.2015.7443332
35. Huang G, Li Y, Wang Q, Ren J, Cheng Y, Zhao X (2019) Automatic classification method for software vulnerability based on deep neural network. IEEE Access 7:28291–28298. https://doi.org/10.1109/ACCESS.2019.2900462
36. Chen J, Kudjo PK, Mensah S, Brown SA, Akorfu G (2020) An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection. J Syst Softw. https://doi.org/10.1016/j.jss.2020.110616
37. Malhotra R, Vidushi (2021) Severity prediction of software vulnerabilities using textual data. International conference on recent trends in machine learning, IoT, Smart Cities, and applications. Advances in Intelligent Systems Computing. https://doi.org/10.1007/978-981-15-7234-0_41
38. Han Z, Li X, Xing Z, Liu H, Feng Z (2017) Learning to predict severity of software vulnerability using only vulnerability description. IEEE Int Conf Softw Maint Evol. https://doi.org/10.1109/ICSME.2017.52
39. Dam HK, Tran T, Pham T, Ng SW, Grundy J, Ghose A (2021) Automatic feature learning for predicting vulnerable software components. IEEE Trans Software Eng 47:67–85. https://doi.org/10.1109/TSE.2018.2881961
40. Russell R, Kim L, Hamilton L, Lazovich T, Harer J, Ozdemir O, Ellingwood P, McConley M (2018) Automated vulnerability detection in source code using deep representation learning. https://doi.org/10.1109/ICMLA.2018.00120.
41. Ban X, Liu S, Chen C, Chua C (2018) A performance evaluation of deep-learnt features for software vulnerability detection. Concurr Comput. https://doi.org/10.1002/cpe.5103
42. Gong X, Xing Z, Li X, Feng Z, Han Z (2019) Joint prediction of multiple vulnerability characteristics through multi-task learning. Int Conf Eng Complex Comput Syst. https://doi.org/10.1109/ICECCS.2019.00011
43. Lin G, Zhang J, Luo W, Pan L, De Vel O, Montague P, Xiang Y (2019) Software vulnerability discovery via learning multi-domain knowledge bases. IEEE Trans Dependable Secure Comput 18(5):2469–2485. https://doi.org/10.1109/TDSC.2019.2954088
44. Liu S, Lin G, Han QL, Wen S, Zhang J, Xiang Y (2019) DeepBalance: deep-learning and fuzzy oversampling for vulnerability detection. IEEE Trans Fuzzy Syst 28(7):1329–1343. https://doi.org/10.1109/TFUZZ.2019.2958558
45. Zhou P, Chen J, Fan M, Du L, Shen YD, Li X (2020) Unsupervised feature selection for balanced clustering. Knowl-Based Syst 193:105417
46. Jiang X, Mao B, Guan J, Huang X (2020) Android malware detection using fine-grained features. Sci Program 2020(1):5190138
47. Zhang X, Wang J, Wang T, Jiang R, Xu J, Zhao L (2021) Robust feature learning for adversarial defense via hierarchical feature alignment. Inf Sci 560:256–270
48. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26(11):1424–1437
49. Mafarja M, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing. https://doi.org/10.1016/j.neucom.2017.04.053

50. Mohammadzadeh H, Gharehchopogh FS (2021) A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: case study email spam detection. Comput Intell. https://doi.org/10.1111/coin.12397
51. Al-Wajih R, Abdulkadir SJ, Aziz N, Al Tashi Q, Talpur N (2021) Hybrid binary grey wolf with Harris hawks optimizer for feature selection. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3060096
52. Singh N, Hachimi H (2018) A new hybrid whale optimizer algorithm with mean strategy of grey wolf optimizer for global optimization. Math Comput Appl. https://doi.org/10.3390/mca23010014
53. Jadhav AN, Gomathi N (2017) Wgc: hybridization of exponential grey wolf optimizer with whale optimization for data clustering. Alex Eng J. https://doi.org/10.1016/j.aej.2017.04.013
54. Jin X, He T, Lin Y (2019) Multi-objective model selection algorithm for online sequential ultimate learning machine. EURASIP J Wirel Commun Netw 2019:1–7
55. Zhang X, Fan M, Wang D, Zhou P, Tao D (2020) Top-k feature selection framework using robust 0–1 integer programming. IEEE Trans Neural Netw Learn Syst 32(7):3005–3019
56. Yaqoob A, Verma NK, Aziz RM, Shah MA (2024) Optimizing cancer classification: a hybrid RDO-XGBoost approach for feature selection and predictive insights. Cancer Immunol Immunother 73(12):261
57. Yaqoob A, Verma NK, Aziz RM, Saxena A (2024) Enhancing feature selection through metaheuristic hybrid cuckoo search and harris hawks optimization for cancer classification. Metaheuristics Mach Learn: Algorithms Appl. https://doi.org/10.1002/9781394233953.ch4
58. Shi B, Chen J, Chen H, Lin W, Yang J, Chen Y, Wu C, Huang Z (2022) Prediction of recurrent spontaneous abortion using evolutionary machine learning with joint self-adaptive sime mould algorithm. Comput Biol Med 148:105885
59. Wolpert DH, Macready WG et al (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
60. Weiss SM, Indurkhya N, Zhang T (2010) Fundamentals of Predictive Text Mining, 1st edn. Springer Publishing Company, Incorporated
61. Willett P (2006) The Porter stemming algorithm: then and now. Program Electron Libr Inf Syst 40(3):219–223. https://doi.org/10.1108/00330330610681295
62. Firpi HA, Goodman E (2004) Swarmed feature selection. Applied imagery pattern recognition workshop 112 Washington DC, USA
63. Abdel-Basset M, El-Shahat D, El-Henawy I, De Albuquerque VHC, Mirjalili S (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Syst Appl 139:112824. https://doi.org/10.1016/j.eswa.2019.112824
64. Tizhoosh HR (2006) Opposition-based reinforcement learning. J Adv Comput Intell Intell Inform 10:578–585. https://doi.org/10.20965/jaciii.2006.p0578
65. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500. https://doi.org/10.1016/j.eswa.2017.07.043
66. Ibrahim RA, Elaziz MA, Lu S (2018) Chaotic opposition-based grey-wolf optimization algorithm based on diferential evolution and disruption operator for global optimization. Expert Syst Appl 108:1–27. https://doi.org/10.1016/j.eswa.2018.04.028
67. He H, Garcia EA (2009) Learning from Imbalanced Data. IEEE Trans Knowl Data Eng 21(9):1263–1284. https://doi.org/10.1109/TKDE.2008.239
68. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
69. He H, Bai Y, Garcia EA., Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. International Joint Conference on Neural Networks
70. Chawla NV (2009) Data mining for imbalanced datasets: an overview. Data mining and knowledge discovery handbook, 875–886
71. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27
72. Lessmann S, Baesens B, Mues C, Pietsch S (2008) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Software Eng 34(4):485–496
73. Olson DL, Delen D (2008) Performance evaluation for predictive modelling. Adv Data Min Tech. https://doi.org/10.1007/978-3-540-76917-0_9
74. Jurafsky D, Martin JH (2020). Speech and Language Processing (3rd ed.). Pearson

75. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press
76. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
77. Harris ZS (1954) Distributional structure. Word 10(2–3):146–162
78. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manage 24(5):513–523
79. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65. https://doi.org/10.1016/j.neucom.2016.03.101
80. Thomas SW, Hassan AE, Blostein D (2014) Mining unstructured software repositories. Evolving Software Systems, Springer, Berlin. https://doi.org/10.1007/978-3-642-45398-4_5
81. Al-Madi N, Faris H, Mirjalili S (2019) Binary multi-verse optimization algorithm for global optimization and discrete problems. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-019-00931-8
82. Emary E, Zawbaa HM (2018) Feature selection via Lèvy Antlion optimization. PAA Pattern Anal Appl. https://doi.org/10.1007/s10044-018-0695-2
83. Sayed GI, Tharwat A, Hassanien AE (2019) Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. Appl Intelligance 49:188–205. https://doi.org/10.1007/s10489-018-1261-8
84. Shi Y, Eberhart RC (1998b) Parameter selection in particle swarm optimization. Evolutionary Programming VII: Proc. EP98. Springer-Verlag, New York:591–600
85. Mitchell M (1998) An introduction to genetic algorithms. MIT press
86. Siedlecki W, Sklansky J (1989) A note on genetic algorithms for large-scale feature selection. Pattern Recognit Lett 10(5):335–347
87. Wohlin C (2007) Empirical software engineering: teaching methods and conducting studies. Empir Softw Eng—Dagstuhl Semin Proc. https://doi.org/10.1007/978-3-540-71301-2_42