Machine Learning Tickets Classification With PSO-Based Feature Selection

Afrah Alharbi, Azzah Alamri, Fatimah Albrahimi, Salma Alyami, Sultan Alqahtani, and Wojdan Alseadan

Computer Science and Information Technology, Imam Muhammad Ibn Saud Islamic Univerisity, Riyadh, Saudi Arabia

{afsaharbi, azhomari, falbrahimi, amasalma}@sm.imamu.edu.sa, ssalqahtani@imamu.edu.sa, wmasaeedan@imamu.edu.sa

***Abstract*—In software projects lifecycle, software maintenance is a critical phase, without this phase software projects would become outdated and inefficient. It is highly important to fix tickets in a short period of time in order to reduce the operation time, reduce cost, and avoid serious losses. To support that, tickets should be correctly and quickly assigned to the right maintenance team by classifying their type. In this paper, PSO is used to implement feature selection over SVM tickets classification with a dataset of 15000 records of pre-labeled tickets. The accuracy was increased from 65.73% to 69% by applying PSO algorithm.**

Keywords—Software Maintenance, Tickets Classification, Machine Learning, Particle Swarm Optimization, Support Vector Machine.

I. Introduction

In software projects lifecycle, software maintenance is a critical phase [1]. Cost and fault effective activities, performed during the pre-delivery stage, as well as the post-delivery stage of a software system are known as software maintenance activities [3]. In the software maintenance phase, customers raise various types of tickets requesting maintainers (organizations or individuals) to modify the software system or its components, correct defects, improve performance, …etc. [2],[3]. Without this phase, software projects would become outdated and inefficient [2].

It is highly important to fix tickets in a short period of time in order to reduce the operation time, reduce cost, and avoid serious losses [1],[3],[4]. To support that, tickets should be correctly and quickly assigned to the right maintenance team by classifying their type.

Determining the type of raised tickets is a part of the software maintenance phase. Machine learning is a strongly useful and important technique for text classification in order to ease the process of information organization and management [5]. In machine learning algorithms every instance in a dataset is represented by using a set of features. The nature of these features could be continuous, categorical, or binary. If instances are provided with label/category then the learning scheme is known as supervised learning. Classification is a supervised machine learning technique that predicts the category of a new input on the basis of training data (Figure 1)[6].

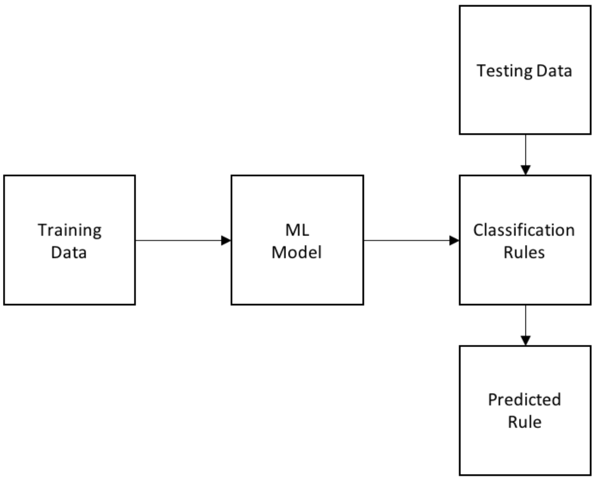

Figure 1 Classification Architecture [6]

Many different text classification techniques have been introduced, including Support Vector Machines (SVM) Decision Trees (DT), K-Nearest Neighbors (KNN), ... etc. [5].

Feature selection techniques are important in pattern classification, data analysis, information retrieval, data processing, machine learning, and data mining applications [7]. In machine learning, feature selection can help as a pre-processing stage of great importance before solving classification problems [7],[8]. The purpose of the feature selection is to reduce the number of irrelevant and noisy features while maintaining an acceptable classification accuracy [7].

Particle swarm optimization (PSO) is an Evolutionary Computation (EC) technique inspired by the behavior of bird predation, proposed by Kennedy and Eberhart in 1995 [9]. PSO purpose is to find the optimal solution through the information sharing and cooperation between individuals in a swarm [8],[9].

In this paper, PSO is used to implement feature selection over SVM tickets classification with a dataset of 15000 records. Tickets are classified into three categories enhancement, bug, and question.

This paper is organized as follows: Section 2. Related Work in tickets and classifications are introduced. Section 3. the proposed method includes PSO, and SVM. Section 4 the experimental results and contains a discussion and comparison with other SVM kernels. Finally, concluding remarks and future work in Section 5.

II. Related Work

Many studies introduced tickets classification with machine learning, the following studies varies in the techniques and results.

Lafi, M. et al [10] classified reported change requests collected from Bugzilla into enhancement, defect, or task. In the preprocessing stage the authors cleaned websites addresses and any non-alphabetic character, the text is converted into lower case after tokenization, they also removed the stop words and lemmatize the text. They finally implement the classification using three different classifiers AdaBoost, Random Forest and Decision Tree. Decision Tree performed slightly better with precision of 78%, recall of 83%, and F1-score of 79%.

Silva, S. & Ribeiro, R. [11] classified incidents in one of the following categories: application, collaboration, enterprise resource planning (ERP), hosting services, network, security and access, output management, software, workplace and support. they performed the pre-processing on incident description starting with the text tokenization, then they eliminated the non-meaningful words based on the resulting dictionary composed by all different words that are present in descriptions. lastly, they applied stemming. After that, all descriptions are represented by a feature vector. They proposed a learning based on SVM and KNN on a real dataset of 10000 incident tickets. they tested their proposed model into three approaches A. Without description (only caller id, severity and the contact source) B. Only description C. All attributes. Approach C where all attributes are used gave slightly better accuracy for both SVM of 89% and KNN of 82%.

Miliano, A. et al [12] used a dataset of 40,251 tickets obtained directly from live Helpdesk ticketing website application. Tickets are categorized based on user's inputted title. The authors used three different machine learning classifiers Decision Tree, Random Forest, and Naive Bayes. In the preprocessing stage the authors removed stop words, numerical characters, punctuation and special characters, and applied word stemming. The results show that Random Forest classification has the highest accuracy value of 82%.

Al-Hawari, F., & Barham, F. [13] used a training data that consists of 1254 technical support pre-labeled tickets. The dataset contains ticket title, description, comments, and reply to comments. Tickets are going to be labeled into one of the following categories: Computers Connections and Parts, Printers, Scanners, Operating Systems, Microsoft Software, Other Software, Other Hardware, Labs Setup, Internet Access, Anti-Virus, Copiers, Check Phone, and Check Projector. The dataset is converted to vectors through StringToWordVector (VSM), then it's been tested on four machine learning techniques: J48 (Tree-based), DecisionTable (Rule based), NaiveBayes (Bayesbased) and SMO (SVM-based). The first experiment using the default feature vectorization parameters with tickets' description only, showed that SMO is giving higher accuracy of 53.80%. The second experiment when the feature vector and Lovins stemmer were enabled to tickets' description only showed that SMO is giving higher accuracy of 59.50%. The last experiment showed the effectiveness of tickets’ description preprocessing by cleaning the dataset from unnecessary HTML tags, punctuation and special characters, and it showed that SMO is giving higher accuracy of 69.90%. Using all tickets parameters (title, description, and comments) gave the optimal result with accuracy of 81.4% for SMO.

Based on our research, supervised machine learning techniques classify tickets with acceptable metrics. The results are varying based on the size of the data used in the training, data preprocessing step, tickets parameters used in the classification, and the type of classifier.

III. Proposed method

In this part the proposed method based on PSO algorithm is introduced to improve the SVM classification. The proposed method is divided into four stages: text pre-processing, text representation, feature selection using PSO, and SVM evaluation.

A. Text Pre-processing

In the text pre-processing stage, the dataset is prepared before passing it to the model in order to maintain informative and clean text, and the text is represented in a proper form that the used model is expecting. The preprocessing stage is divided into three steps: data collection, data cleaning, and tokenization.

A1. Data Collection

Dataset consists of 15000 gathered GitHub issues (in English). Issues are pre-labeled into three categories: bug, enhancement, and question. Labels are represented numerically, 0 for issues, 1 for enhancement, and 2 for question. Lastly, for each issue title and description are concatenated into one single text.

A2. Data Cleaning

Since the dataset is collected from users directly, it requires cleaning as it contains non-informative data. In this step emojis, symbols, pictographs, non-English characters, numbers, and website links are eliminated from all issues.

A3. Tokenization

Tokenization is the manner of breaking up text into pieces (words) called tokens. The collection of tokens is then used for further processing [14]. After data cleaning is done, issues are tokenized into words.

B. Text Representation

After the text pre-preprocessing is finished, all issues are represented by a feature vector using fastText. fastText is a library created by Facebook's AI Research lab for word embedding and text classification [15]. The fastText can find vector representations for words that are not directly found in the dictionary [16]. In this step, fastText library is used to generate word embedding for tokens (words). Each token in an issue has 300 generated vectors representation using fastText. For all tokens in one issue, the average of vectors is calculated to generate 300 vectors for the entire issue (Figure 2). A new dataset is created that contains each issue representation in vectors, assigned to its label. Vectors are treated as features to be used later in the model.

C. Feature Selection with PSO

Feature selection refers to the keeping most useful features from a dataset. In this stage, PSO is used to select features that improve the quality of a classification model. PSO represents the social behavior of the swarm, which refers to the set of solutions where each solution is a particle. The PSO algorithm uses the global best solution to obtain the optimal solution. in each PSO iteration, the global best solution is recorded and updated if it met the condition [17]. The pseudocode of the PSO is indicated below (Algorithm 1).

```
Input: Generate the initial particles randomly.
Output: Optimal particle and its fitness value.
Algorithm
Initialize swarm and parameters of the particle swarm optimization c1,
c2,W and N.          //N is the number of iterations
Evaluate all particles using the fitness function by Eq. (1).
while i < N  do
  Update the velocity
  Update each position
  Evaluate the fitness function.
  Replaces the worst particle with best particle.
  Update LB and GB.
end while
Return a subset of selected features.
```

Algorithm 1 PSO Algorithm [17]

The dataset of features is passed to the PSO algorithm in order to reduce the number of features, from 300 features to an optimal subset of features. First, the PSO parameters are initialized as follow (table1):

Table 1 PSO Parameters

PSO Parameter	Description
N	Number of Iteration
C1, C2	Constriction Coefficient
W	Inertia Weight

In order to update each particle in PSO, two main factors are used: particle position (Equation 1), and the velocity (Equation 2). updating the velocity is based on the movement effect of a particle, where each particle attempts to move to the optimal position [17].

$$x_{ij} = x_{ij} + v_{ij} \qquad \text{(Equation 1)}$$

$$v_{ij} = w * \times v_{ij} + c1 \times rand1 \times (LB_{l} - x_{ij}) + c2 \times rand2 \times (GB_{l} - x_{ij}) \qquad \text{(Equation 2)}$$

each parameter in the equations is described as follow (Table 2).

Table 2 Velocity Parameters

PSO Parameter	Description
LB	Current best local solution
GB	Current best global solution
rand1, rand2	Random numbers in the range of [0, 1]
V	Particle's velocity
X	Particle's position

The fitness function is an evaluation measure used to evaluate each candidate solutions. If the quality of the solution is increased, the solution will be updated. The solution with high fitness function value is considered to be the optimal [18].

The fitness function depends on the accuracy of the SVM model. The fitness function is calculated using the equation below (Equation 3) [18].

$$f(\boldsymbol{u}_i) = \alpha(1-P) + (1-\alpha)\left(1 - \frac{N_f}{N_t}\right), \qquad \text{(Equation 3)}$$

The fitness function parameters are described as follow (Table 3):

Table 3 Fitness Function Parameters

PSO Parameter	Description
LB	Current best local solution
GB	Current best global solution
rand1, rand2	Random numbers in the range of [0, 1]
V	Particle's velocity
X	Particle's position

At the final iteration, the optimal subset of feature is returned to be used in the next stage.

D. SVM Evaluation

After the PSO preforms the feature selection, the selected features will be passed to the SVM model. SVM model is used for issues classification, it has been evaluated through four metrices: accuracy, precision, recall, and F1-measure.

The following flowchart summarizes the model used in issues classification (Figure 2).

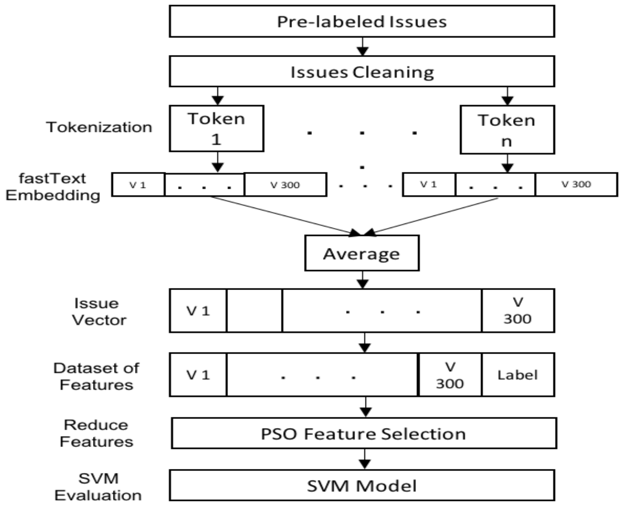

Figure 2 Proposed Model for Issues Classification

IV. Results and Discussion

Experiment were tested on a subset of public dataset[1] containing 15,000 pre-labeled issues into one of three categories: bug, enhancement, and question. The dataset was collected by querying the GitHub Archive[2] using Google BigQuery[3].

In the text preprocessing stage, the dataset has been tested on two approaches: in Approach.1 the dataset after cleaning (discussed in section III), in Approach.2 the dataset with stemming and stop words removal. Approach.1 showed higher classification accuracy of 65.73% using SVM, and the accuracy of Approach.2 was 60.9%.

The SVM model was tested on two kernels: RBF, and Linear. We selected the RBF kernel as it showed higher accuracy of 65.73%, while the Linear kernel showed an accuracy of 64.4%.

We compared SVM with RBF kernel to other machine learning technique named Logistic Regression. SVM showed higher accuracy of 65.73%, while in Logistic Regression the accuracy was 64%. Based on the above experiments we selected SVM with RBF kernel to be applied in our model.

After selecting the proper machine leaning model, we applied PSO to the dataset to implement feature selection over SVM tickets classification. The following experiments are done to select PSO parameters to achieve high performance.

First Experiment:

In the first experiment we tested our model by changing the number of particles, while other parameters were kept constant as follow, c1=0, c2=4.1, w=0.1. The number of particles is selected based on the accuracy (Table 4). Figure 3 illustrates the metrices when changing the number of particles.

Table 4 First Experiment

Number of particles	Accuracy	Precision	Recall	F-1
10	64.67%	64.75%	64.86%	64.64%
30	66%	65.96%	66.11%	65.88%
60	65%	65.08%	65.09%	64.98%
100	66.33%	66.29%	66.46%	66.32%
150	65.67%	65.66%	65.84%	65.63%

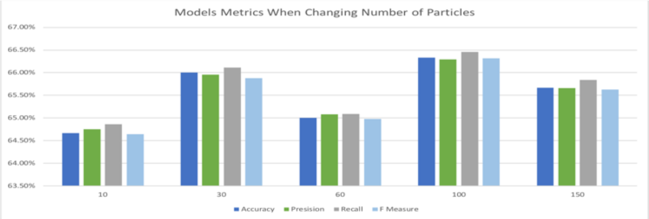

Figure 3 Experiment 1 Illustration

Considering the first experiment, a higher accuracy of 66.33% when the number of particles is 100, and 185 features been selected. Based on that, 100 particles are going to be used in next experiments.

Second Experiment:

In the second experiment we tested our model by changing c1, and c2, while other parameters were kept constant as follow, number of particles= 100, and w=0.1. c1, and c2 are selected based on the accuracy (Table 5). Figure 4 illustrates the metrices when changing c1 and c2.

Table 5 Second Experiment

C1	C2	Accuracy	Precision	Recall	F-1
0.5	3.6	64%	64%	64.09%	63.99%
1.5	2.6	66.33%	66.44%	66.44%	66.44%
3.5	0.6	65%	65.02%	65.28%	64.96%
2.5	1.6	69%	68.94%	69.11%	68.86%
4.1	0	66%	66.50%	66%	66%

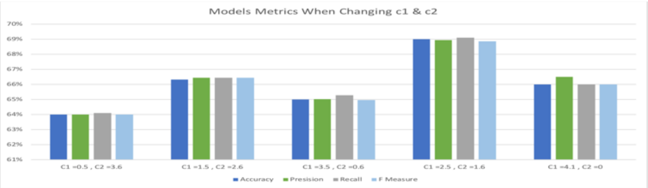

Figure 4 Experiment 2 Illustration

Considering the second experiment, a higher accuracy of 69% when the number of particles is 100, c1=2.5, c2=1.6, and 172 features been selected. Based on that, PSO parameters are set to 100 particles, c1= 2.5 and c2=1.6 to be used in next experiments.

Third Experiment:

In the third experiment we tested our model by changing the inertia weight w, while other parameters were kept constant as follow, number of particles = 100, c1=2.5, and c2=1.6. The inertia weight w is selected based on the accuracy (Table 6). Figure 5 illustrates the metrices when changing inertia weight w.

Table 6 Third Experiment

w	Accuracy	Precision	Recall	F-1
0.3	64.66%	64.70%	64.78%	64.70%
0.5	67.33%	67.29%	67.42%	67.31%
0.7	64.66%	64.60%	65.69%	64.50%
0.9	64.66%	64.72%	64.74%	64.63%
1	66.30%	66.31%	66.33%	66.24%

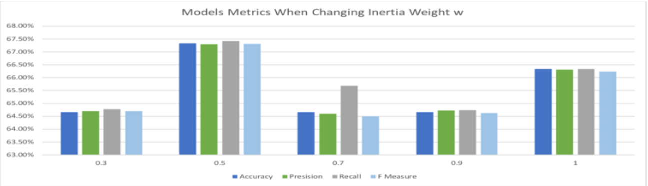

Figure 5 Experiment 3 Illustration

[1] https://www.kaggle.com/davidshinn/github-issues?select=github_issues.csv

[2] https://gharchive.org

[3] https://cloud.google.com/bigquery

Considering all experiment, a higher accuracy of 69 % when the number of particles is 100, c1=2.5, c2=1.6, and w=0.1, and 172 features been selected.

Based on that, the PSO improved the classification accuracy using SVM with RBF kernel. The accuracy was increased from 65.73% to 69%.

V. Conclusion

In software projects lifecycle, without software maintenance projects would become outdated and inefficient. It is highly important to fix tickets quickly in order to reduce the operation time, reduce cost, and avoid serious losses. To support that we proposed SVM issues classification model that classifies issues into one of three categories: bug, enhancement, and question. PSO is used to implement feature selection over SVM to improve the model performance. The model was tested on a dataset of 15000 records of pre-labeled issues. The accuracy was increased from 65.73% to 69% by applying PSO algorithm. In future, deep learning algorithms will be applied in the classification in order to enhance the performance of the classification.

References

[1] Di Lucca, G. A., Di Penta, M., & Gradara, S. (2002, October). An approach to classify software maintenance requests. In International Conference on Software Maintenance, 2002. Proceedings. (pp. 93-102). IEEE.

[2] L'Erario, A., Thomazinho, H. C. S., & Fabri, J. A. (2020). An approach to software maintenance: A case study in small and medium-sized businesses it organizations. International Journal of Software Engineering and Knowledge Engineering, 30(05), 603-630.

[3] Mahmoodian, N., Abdullah, R., & Murad, M. A. A. (2010, March). A framework of classifying maintenance requests based on learning techniques. In 2010 International Conference on Information Retrieval & Knowledge Management (CAMP) (pp. 245-249). IEEE.

[4] Kallis, R., Di Sorbo, A., Canfora, G., & Panichella, S. (2019, September). Ticket tagger: Machine learning driven issue classification. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 406-409). IEEE.

[5] Zhou, X., Gururajan, R., Li, Y., Venkataraman, R., Tao, X., Bargshady, G., ... & Kondalsamy-Chennakesavan, S. (2020, August). A survey on text classification and its applications. In Web Intelligence (No. Preprint, pp. 1-12). IOS Press.

[6] P. C. Sahu, S. K. Rauta, and D. R. Dash, 'An Approach to Supervised Machine Learning: A survey', p. 7.

[7] Tu, C.-J., Chuang, L.-Y., Chang, J.-Y., Yang, C.-H., & Others. (2007). Feature selection using PSO-SVM. International Journal of Computer Science.

[8] Bai, X., Gao, X., & Xue, B. (2018, July). Particle swarm optimization based two-stage feature selection in text mining. In 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.

[10] Lafi, M., Hawashin, B., & AlZu'bi, S. (2020, April). Maintenance Requests Labeling Using Machine Learning Classification. In 2020 Seventh International Conference on Software Defined Systems (SDS) (pp. 245-249). IEEE.

[11] Silva, S., Pereira, R., & Ribeiro, R. (2018, June). Machine learning in incident categorization automation. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

[12] Miliano, A., Steven, I., Kosim, K. P., Jayadi, R., & Mauritsius, T. (2020, December). Machine Learning-based Automated Problem Categorization in a Helpdesk Ticketing Application. In 2020 8th International Conference on Orange Technology (ICOT) (pp. 1-6). IEEE.

[13] Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. Journal of King Saud University-Computer and Information Sciences, 33(6), 702-718.

[14] Kilimci, Z. H., &amp; Akyokus, S. (2018). Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification. Complexity, 2018.

[15] Ritu, Z. S., Nowshin, N., Nahid, M. M. H., & Ismail, S. (2018, September). Performance analysis of different word embedding models on bangla language. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) (pp. 1-5). IEEE.

[16] Kuyumcu, B., Aksakalli, C., &amp; Delil, S. (2019, June). An automated new approach in fast text classification (fastText) A case study for Turkish text classification without pre-processing. In Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval (pp. 1-4).

[17] [6:24 PM] Afrah Saleh Jazi Alharbi Abualigah, L. M., Khader, A. T., &amp; Hanandeh, E. S. (2018). A new feature selection method to improve thedocument clustering using particle swarm optimization algorithm. Journal of Computational Science, 25, 456-466.

[18] Vieira, S. M., Mendonça, L. F., Farinha, G. J., & Sousa, J. M. (2013). Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. Applied Soft Computing, 13(8), 3494-3504.