ICIC Express Letters
Volume **16**, Number **5**, May **2022**

 ISSN 1881-803X
pp. **449–457**

OPTIMIZATION LASSO-SUPPORT VECTOR MACHINE ON SOFTWARE DEFECT PREDICTION

Anang Prasetyo[1], Antoni Wibowo[2] and Diana[2]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
[2]Computer Science Department, School of Computer Science
Bina Nusantara University
Jl. Kebun Jeruk Raya No. 27, Jakarta 11530, Indonesia
anang.prasetyo@binus.ac.id; {anwibowo; diana}@binus.edu

Received June 2021; accepted September 2021

Abstract. *Software defect has now become the most important concern in a recent study. This is because the software has become an inseparable part of today's modern era, and software as it evolves becomes increasingly complex. Data complexity is one of the problems in modeling software defect prediction, so the previous research proposed the Least Absolute Shrinkage and Selection Operator (LASSO) algorithm to reduce data with the Support Vector Machine (SVM) model called LASSO-SVM, but from the study, there are still not optimal parameters selected. To solve the problem this research proposed the addition of Grid Search to the modeling of LASSO-SVM. Grid Search will be used to find the optimal parameters for the SVM model. Finally results of the experiment of Tune LASSO-SVM are 6% higher accuracy, 7% precision, 9% recall, and 5% F1 score when compared to previous LASSO-SVM models. So Tune LASSO-SVM obtained a better and stable model in terms of predictions compared to previous LASSO-SVM models.*
Keywords: Software defect prediction, Support vector machine, Feature selection, Grid Search, Cross-validation

1. **Introduction.** Nowadays software is very useful and commonly used in the modern era. This is because software is widely used in various fields, such as banking systems, biopharmaceutical engineering, and traffic systems [1]. In recent years the advancement of software systems has been increasingly advanced and reliable [2]. Because the many areas that develop software systems result in software now becoming complex, it is very important to identify and repair any software defects [3].

A software defect is a necessary part of improving the quality of the software and reducing development costs [4]. Therefore, software defect predictions are proposed to assist the team in finding the presence of corrupted code more easily [5]. Software defect prediction is also much more effective for detecting software defects when compared to software testing and reviews [6]. Defect prediction includes part of active research in software engineering [7]. At the beginning of the development of software defect prediction there has been developed several prediction techniques ranging from data mining techniques and machine learning techniques [8].

In a recent study on software defect prediction, many researchers proposed some algorithms to improve modeling. Shuai et al. [9] raised the issue of traditional SVM classification on software defect prediction by proposing Cost-Sensitive Support Vector Machine (CSSVM) and Genetic Algorithm (GA) models. The GA-CSSVM model will use GA optimization to find the optimal solution. Li et al. [10] raised the topic of software defect prediction based on nature, size of the software, complexity, and development methods using Fuzzy Measure (FM) based on Genetic Algorithm (GA). Shan et al. [11] raised the issue of data redundancy due to the large number of attributes on the dataset used

DOI: 10.24507/icicel.16.05.449

which resulted in a decrease in accuracy, using the Locally Linear Embedding and Support Vector Machine (LLE-SVM) models.

Li et al. [12] raised the issue of Support Vector Machine (SVM) modeling in determining optimal parameters in software defect prediction by proposing Change Range Bat Algorithm (CRBA) and Support Vector Machine or namely CRBA-SVM models. The CRBA-SVM for short use modified bat algorithms to optimize SVM parameters. Shan et al. [13] raised the issue of lower model efficiency and lower predictive accuracy of LLE-SVM and proposed Improvement Locally Linear Embedding and Support Vector Machine (ILLE-SVM). The ILLE-SVM model uses a Grid Search algorithm to search for optimal parameters, reducing the time of optimizing parameters gradually. Gan and Zhang [14] raised the issue of accuracy and application of most traditional software defect predictions that are not very high by proposing a Grey Relational Analysis and Support Vector Machine (GRA-SVM) model. GRA algorithms play a role in reducing the dimensions of software metrics and curating irrelevant data and the use of SVM models for software prediction defect modeling. Muthukumaran et al. [15] raised the issue of assumption the existence of conditional independence affects in the results of classifying, using the method of Random Forest and Naïve Bayes. The result of the evaluation obtained random forest classification is better than using Naïve Bayes.

Wei et al. [16] raised the issue of software defect predictions on the redundancy of data caused by multi-dimensional measurements and led to a decrease in prediction accuracy with Neighborhood Preserving Embedding and Support Vector Machine (NPE-SVM) modeling solutions. NPE will be used to keep the structure of the data unchanged during the data dimension reduction process, followed by SVM modeling. Cai et al. [1] raised the issue of class imbalance and parameter selection in Support Vector Machine (SVM) modeling in software defect prediction by proposing the Under Sample Hybrid Multi-Objective Cuckoo Search (HMOCS-US-SVM) method. Wang et al. [18] raised the issue of poor predictive accuracy of most existing software defect prediction models by proposing the LASSO-SVM model. LASSO algorithms are used to reduce the dimensions of the original data and remove unnecessary data, and then combined with cross-validation algorithms to determine optimal parameters in Support Vector Machine (SVM) modeling.

This study using the previous software defect prediction studies of LASSO-SVM modeling techniques from Wang et al. [18], with the addition of the Least Absolute Shrinkage and Selection Operator (LASSO) algorithm, obtained accuracy results of 92.14%, precision level of 66.67%, recall rate of 78.04%, and F1 score of 72.72% in PC1 dataset. Based on the results of previous research, it is known that SVM modeling with the addition of the LASSO-SVM has weaknesses in the selection of optimal parameters, so the prediction results are not consistent. Finally, the LASSO algorithm keeps use in this study to overcome data redundancy by looking for the best features. Furthermore, to overcome the problem of finding optimal parameters, it is proposed to add a Grid Search algorithm. Grid Search algorithm will be used to find optimal parameters for the SVM model. For evaluating the proposed method, this study will test using NASA public dataset CM1, JM1, KC1 and PC1 datasets from PROMISE repository [20].

Section 1 explains the background of research, related research and motivation of research conducted. Section 2 describes the literature of research studies. The proposed methods and application of algorithms are illustrated in Section 3. Section 4 describes the experiments conducted and measurement techniques in the study. Finally conclusion results of the experiment and plans for the next research are summarized in Section 5.

2. **Related Work.**

2.1. **Software defect prediction.**

Defect prediction is one of the researches that is still active today in the field of software engineering [21]. Some empirical studies show that

software defect is not uniformly distributed in software modules but rather only a few, covering a large amount of temporary defect, some even no defect at all [22]. Analyzing and predicting defects to software is necessary to achieve three main objectives [23]: first to assess the progress of a developed project and plan activities to detect defects for project managers, second to evaluate the quality of products developed and the last to assess the performance of the developed project management.

Wahono [6] studied that many companies develop defect prediction models using company-owned data and present models in conferences. However, this is not used as a reference, because the dataset used cannot be assessed as a standard dataset for software defect prediction. In the last 10 years, hundreds of different defect prediction models have been developed and published [24], with a predicted classification performance of about 80%. Based on the research that has been done in software defect prediction obtaining mixed results, several methods can detect classifying with small data dimensions, and some methods have a deficiency inconsistency in software defect prediction.

2.2. **Support vector machine.** Support Vector Machine (SVM) is one of the modeling methods that can compress classification and regression [25]. The strength of the SVM algorithm is because it can perform data classification patterns and accuracy in a balanced manner. SVM has been widely used by researchers as a tool for modeling in terms of classification and has been used several times to solve regression problems in several scenarios. The main goal of SVM is to find the optimal hyperplane that separates the training points into two classes with the maximal margin, and also provides way to deal with outlier by finding the optimal margins of each class [26, 27]. The schematic diagram of SVM can be seen in Figure 1.

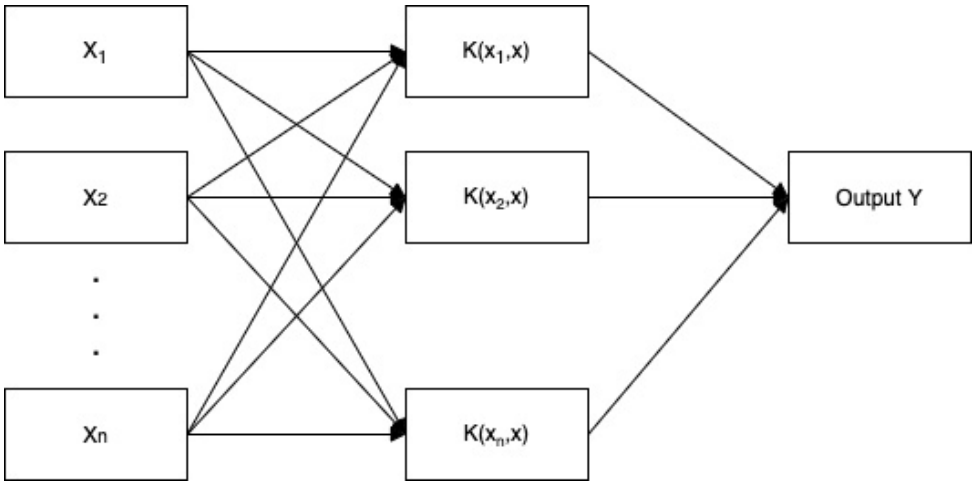

FIGURE 1. Semantic diagram of SVM

2.3. **Least absolute shrinkage and selection operator.** The Least Absolute Shrinkage and Selection Operator (LASSO) was proposed by Tibshirani [17] for parameter estimation and also variable (model) selection simultaneously in regression analysis. LASSO can shrink the regression coefficient of a less important variable to 0 with a penalty function [18]. The LASSO estimate can be defined as Equation (1).

$$\sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1}$$

In Software Defect Prediction (SDP) LASSO can be used to reduce variance and minimize bias which is a common error in model predictions. LASSO is also a useful tool to minimize variables irrelevant for modeling. LASSO changes each coefficient with a

constant component λ, to zero. The larger λ is, the more coefficient is converted to zero [19].

2.4. **Grid Search and cross-validation.** Grid Search is a complete search based on a specified subset of hyper-parameter spaces [28]. Due to complex computing, Grid Search is only suitable for fewer parameters [29]. Grid Search is one of the methods widely used in SVM modeling [30]. The basic process of Grid Search is to set the value interval and step length of the parameter and adjust the C value, γ at each interval. The value (C, γ) will form a grid and will be calculated of the highest accuracy value with cross-validation technique, and then select the highest accuracy value as the best parameter.

3. **Methodology.** In this study, the design of the proposed methodology is diagrammatically presented in Figure 2. The methodology separates in some steps including initialize, preprocessing data, modelling & optimization, and result. This study uses datasets from NASA datasets of PROMISE repository namely CM1, JM1, KC1, and PC1 which were selected to classify models for software defect prediction and more in-depth evaluation.

The datasets that were collected will be cleansing first to ensure that the data is ready for use in modeling by checking no data is empty or has values that do not match the data type. In the experiment, cleansing data consist of deleting data that have inappropriate

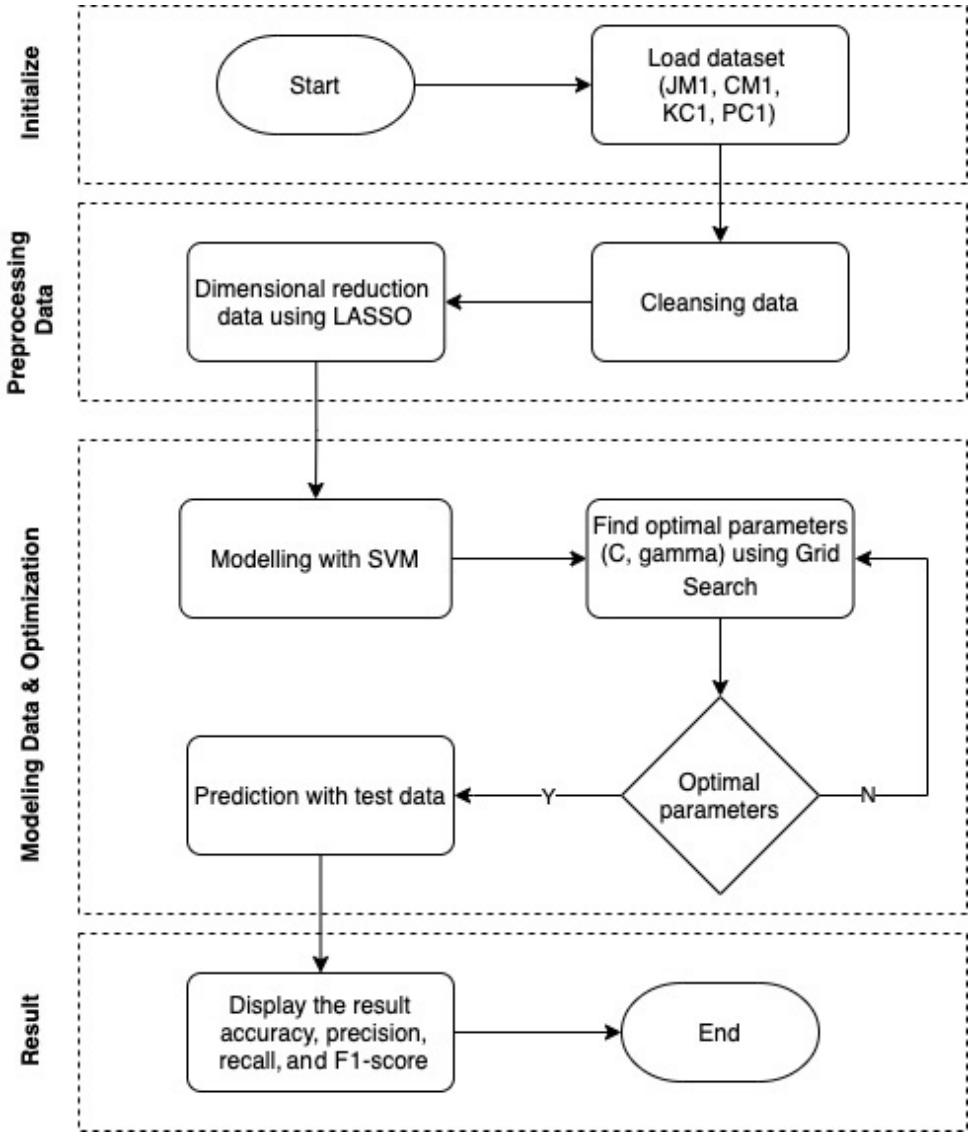

FIGURE 2. Design experiment

values, removing duplicate data, and equating data types for each column. After process cleansing data will be fed as input to the feature selection process using LASSO algorithm. LASSO algorithm will eliminate the feature with low correlation with class and select the high scores feature correlation with class. The selected feature is used to classify into two classes, defect and non-defect software, using SVM model. The SVM modelling will find the optimal hyperplane that separates the training points into two classes with the maximal margin. To enhance the accuracy of the existing classifier use Grid Search algorithm for hyperparameter tunning. The basic concept of grid search algorithm will be set of the value interval from step length of the parameter (C, γ) and adjust the C value, γ at each interval using cross-validation technique to find the combination of parameter with the highest accuracy value. The performance evaluation is carried out to evaluate performance of the model proposed to predict the software defect or not defect. The measurement matrix used in this study is accuracy, precision, recall, and F1 score.

4. **Experiments.**

4.1. **Experimental environment and environment dataset.**

The implementation in this study used Jupyter Notebook with Python programming language. The program will run on a computer with a 1.6GHz Dual-Core Intel Core i5 processor and 4GHz memory. Environment data used in this study include CM1, JM1, KC1, and PC1 datasets from the NASA PROMISE repository. Details of each dataset can be seen in Table 1.

Table 1. Dataset details

Dataset name	Module	Defect	Defect (%)
CM1	498	49	9.83%
JM1	10885	2106	19.35%
KC1	2109	326	15.45%
PC1	1109	77	6.94%

4.2. **Experimental data preprocessing.**

The NASA dataset used in this study has several problems that must be resolved before the modeling process. One of them is the number of duplicate data in the dataset and the data class between the data defect and not too far, this greatly affects the performance of the predicted software defects. In this paper, the first thing to do is to remove the duplication of data from all datasets CM1, JM1, KC1, and PC1. Next is to normalize the data using the minimum and maximum methods of normalization, where each value of the attribute is changed to a value with a range of $(0, 1)$.

Finally, the LASSO method is used to reduce the data dimensions from the results of data normalization that has been done. After performing feature reduction using LASSO, the attributes that affect the result are eliminated. The attributes that are most important will be used in the SVM modeling process. In the modeling process, hyperparameter tuning will be carried out using Grid Search and cross-validation to get optimal parameter values. The last result of the model's performance is accuracy, precision, recall, and F1 score.

4.3. **Experimental evaluation.**

In this paper, evaluation techniques based on the confusion matrix are used to predict the performance of the predictive model developed [31]. Confusion matrix generally consists of two classes, namely actual class and predicted class [32]. The general model of confusion matrix can be seen in Table 2.

Finally, the measurement metrics used are accuracy, precision, recall, and F1 score values that can be seen in Table 3.

Table 2. Confusion matrix

Actual class	Predicted class	
	Minimum	Maximum
Defective	True Positive (TP)	False Negative (FN)
Non defective	False Positive (FP)	True Negative (TN)

Table 3. Metric measurement for software defect prediction

Metric	Formula	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall model performance value
Precision	$\frac{TP}{TP+FP}$	How accurately predictions are positive
Recall	$\frac{TP}{TP+FN}$	Actual scope of positive data examples
F1 score	$\frac{2TP}{2TP+FP+FN}$	Mixed metrics used for disproportionate classes

4.4. **Analysis of experimental results.** In experiments use accuracy values in a tenfold cross-validation algorithm, where the data is divided into ten parts, nine for training and one for test data combined with the Grid Search algorithm. The process will be repeated ten times with a combination of the specified C and γ parameters, and then it will be calculated as the average value and become the final value. Some tests conducted on several datasets CM1, JM1, KC1, and PC1 producing different values can be seen in Tables 4-7, in terms of modeling, also set class weight in SVM model that aims to balance the class on each dataset. Optimizations performed on the LASSO-SVM model show better results where the value of the metrics measurement is more stable than the previous models.

In the CM1 dataset Tune LASSO-SVM in Table 4 scored better accuracy, recall, and F1 score than the other models, but lower precision values than both of models. So the accurate prediction positive value is not better.

Table 4. Result of CM1 dataset

Predictive model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
SVM	73.02	85.9	73.02	77.42
LASSO-SVM	72.77	86.84	72.77	77.35
Tune LASSO-SVM	82.72	82.72	82.72	82.72

In JM1 dataset Tune LASSO-SVM in Table 5 gets a better accuracy, precision, recall value than the other models, while the F1 score is lower, this is because the JM1 dataset has a considerable imbalanced data problem, so the performed prediction model is impacted.

Table 5. Result of JM1 dataset

Predictive model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
SVM	67.64	72.61	67.64	69.42
LASSO-SVM	69.58	72.62	69.58	70.8
Tune LASSO-SVM	76.98	79.98	76.98	67.55

In the KC1 dataset Tune LASSO-SVM in Table 6 scored better accuracy, precision, recall, and F1 score than the other models. So the performance of the proposed method is better for software defect prediction.

Table 6. Result of KC1 dataset

Predictive model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
SVM	67.56	66.35	67.56	66.86
LASSO-SVM	65.93	65.23	65.93	65.55
Tune LASSO-SVM	74.52	70.22	74.52	69.99

PC1 dataset in Table 7 served as a reference for previous research, compared with the previous research the Tune LASSO-SVM gets a better precision value, recall, and F1 score, although the accuracy value is slightly decreased, on a model basis it is more stable in terms of measurement of the other three measurement metrics.

Table 7. Result of PC1 dataset

Predictive model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
SVM	78.26	79.4	75.46	79.85
LASSO-SVM	92.14	65.6	77.9	71.6
Tune LASSO-SVM	91.38	85.54	91.38	88.36

In addition, for the average performance result of datasets the Tune LASSO-SVM has 6% higher accuracy, 7% precision, 9% recall, and 5% F1 score when compared to previous LASSO-SVM models. From these results, it can be known that the performance of the Tune LASSO-SVM model can predict the defect module better. Finally, it is known that Tune LASSO-SVM results using Grid Search algorithms can get optimal parameters, in addition to the LASSO algorithm that helps reduce data dimensions to speed up the modeling process.

5. **Conclusions.** From the results of experiments that have been conducted, it was found that the optimization LASSO-SVM model can solve the problem of optimal parameter selection on the model. This is shown in the performance value of the model which is much more stable and better compared with the other models, traditional SVM and LASSO SVM previously, on each of the datasets tested.

However, in some datasets, the accuracy value is lower than other models that occur in PC1 dataset and in dataset CM1, KC1 with LASSO-SVM model without hyperparameter tuning results is the performance lower than traditional SVM. So from the results obtained LASSO algorithm is not suitable for use in CM1 and KC1 datasets, and to overcome this will be the target of further research.

Acknowledgment. This work is partially supported by Bina Nusantara University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Cai, Y. Niu, S. Geng, J. Zhang, Z. Cui, J. Li and J. Chen, An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search, *Concurrency and Computation: Practice and Experience*, vol.32, no.5, pp.1-14, 2020.

[2] X. Huo and M. Li, On cost-effective software defect prediction: Classification or ranking?, *Neurocomputing*, vol.363, pp.339-350, 2019.

[3] P. Paramshetti and D. Phalke, Survey on software defect prediction using machine learning techniques, *Journal of Applied Mathematics*, vol.3, no.12, 2014.

[4] A. Majd, M. Vahidi-Asl, A. Khalilian, P. Poorsarvi-Tehrani and H. Haghighi, SLDeep: Statement-level defect prediction software using deep-learning model on static code features, *Expert Systems with Applications*, vol.147, 2020.

[5] J. Deng, L. Lu and S. Qiu, Software defect prediction via LSTM, *IET Software*, vol.14, no.4, pp.443-450, 2020.

[6] R. S. Wahono, A systematic literature review of software defect prediction: Research trends, datasets, methods and frameworks, *Journal of Software Engineering*, vol.1, pp.1-16, 2015.
[7] L. Qiao, X. Li, Q. Umer and P. Guo, Deep learning based software defect prediction, *Neurocomputing*, vol.385, pp.100-110, 2020.
[8] C. Manjula and L. Florence, Deep neural network based hybrid approach for software defect prediction using software metrics, *Cluster Computing*, vol.22, pp.9847-9863, 2019.
[9] B. Shuai, H. Li, M. Li, Q. Zhang and C. Tang, Defect prediction software using dynamic support vector machine, *Proc. of the 9th International Conference on Computational Intelligence and Security (CIS2013)*, pp.260-263, 2013.
[10] K. Li, C. Chen, W. Liu, X. Fang and Q. Lu, Software defect prediction using fuzzy integral fusion based on GA-FM, *Wuhan University Journal of Natural Sciences*, vol.19, no.5, pp.405-408, 2014.
[11] C. Shan, B. Chen, C. Hu, J. Xue and N. Li, Software defect prediction model based on LLE and SVM, *IET Conference Publications*, pp.1-5, 2014.
[12] F. Li, X. Rong and Z. Cui, A hybrid CRBA-SVM model for software defect prediction, *International Journal of Wireless and Mobile Computing*, vol.10, no.2, pp.191-196, 2016.
[13] C. Shan, H. Zhu, C. Hu, J. Cui and J. Xue, Software defect prediction model based on improved LLE-SVM, *Proc. of 2015 4th International Conference on Computer Science and Network Technology*, pp.530-535, 2015.
[14] Y. Gan and C. Zhang, Research of software defect prediction based on GRA-SVM, *AIP Conference Proceedings*, 2017.
[15] K. Muthukumaran, S. Srinivas, A. Malapati and L. B. M. Neti, Software defect prediction using augmented Bayesian networks, in *Proceedings of the 8th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016). SoCPaR 2016. Advances in Intelligent Systems and Computing*, A. Abraham, A. Cherukuri, A. Madureira and A. Muda (eds.), Cham, Springer, 2018.
[16] H. Wei, C. Shan, C. Hu, H. Sun and M. Lei, Software defect distribution prediction model based on NPE-SVM, *China Communications*, vol.15, no.5, pp.173-182, 2018.
[17] R. Tibshirani, Regression shrinkage and selection via the LASSO, *Journal of the Royal Statistical Society, Series B*, vol.58, no.1, pp.267-288, 1996.
[18] K. Wang, L. Liu, C. Yuan and Z. Wang, Software defect prediction model based on LASSO-SVM, *Neural Computing and Applications*, vol.33, no.14, pp.8249-8259, 2020.
[19] R. Muthukrishnan and R. Rohini, LASSO: A feature selection technique in predictive modeling for machine learning, *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pp.18-20, 2016.
[20] S. Sayyad and T. Menzies, *The PROMISE Repository of Software Engineering Databases*, University of Ottowa, Canada, 2015.
[21] S. Omri and C. Sinz, Deep learning for software defect prediction: A survey, *Proc. of 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW2020)*, pp.209-214, 2020.
[22] Z. Yan, X. Chen and P. Guo, Software defect prediction using fuzzy support vector regression, in *Advances in Neural Networks – ISNN 2010. ISNN 2010. Lecture Notes in Computer Science*, L. Zhang, B.-L. Lu and J. Kwok (eds.), Berlin, Heidelberg, Springer, 2010.
[23] G. Abaei and A. Congratulations, A survey on fault detection software based on different prediction approaches, *Vietnam Journal of Computer Science*, vol.1, no.2, pp.79-95, 2014.
[24] D. Bowes, T. Hall and J. Petrić, Software defect prediction: Do different classifiers find the same defects, *Software Quality Journal*, vol.26, no.2, pp.525-552, 2018.
[25] D. A. Pisner and D. M. Schnyer, *Support Vector Machine*, Power Systems Elsevier, Inc., 2007.
[26] H. Dyoniputri and Afiahayati, A hybrid convolutional neural network and support vector machine for dysarthria speech classification, *International Journal of Innovative Computing, Information and Control*, vol.17, no.1, pp.111-123, 2021.
[27] L. Zhong and T. Wang, Towards word sense disambiguation using multiple kernel support vector machine, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.555-570, 2020.
[28] I. Syarif, A. Prugel-Bennett and G. Wills, SVM parameter optimization using grid search and genetic algorithm to improve classification performance, *Telkomnika (Telecommunication Computing Electronics and Control)*, vol.14, no.4, pp.1502-1509, 2016.
[29] F. Friedrichs and C. Igel, Evolutionary tuning of multiple SVM parameters, *Neurocomputing*, vol.64, nos.1-4, pp.107-117, 2005.
[30] H. Wei, C. Hu, S. Chen, Y. Xue and Q. Zhang, Establishing a software defect prediction model via effective dimension reduction, *Information Sciences*, vol.477, pp.399-409, 2019.

[31] W. Rhmann, Application of hybrid search based algorithms for software defect prediction, *International Journal of Modern Education and Computer Science*, vol.10, no.4, pp.51-62, 2018.

[32] R. Jayanthi and L. Florence, Software defect prediction techniques using metrics based on neural network classifier, *Cluster Computing*, vol.22, pp.77-88, 2019.