



Detection of Web Service Anti-patterns Using Neural Networks with Multiple Layers

Sahithi Tummalapalli¹, Lov kumar²(✉), N. L. Bhanu Murthy¹,
and Aneesh Krishna²

¹ BITS Pilani Hyderabad, Hyderabad, India
{P20170433,lovkumar,bhanu}@hyderabad.bitspilani.ac.in

² Curtin University, Perth, Australia
a.krishna@curtin.edu.au

Abstract. Anti-patterns in service-oriented architecture are solutions to common issues where the solution is ineffective and may end up in undesired consequences. It is a standard exercise that initially seems like the best solution; however, it finally ends up having bad results that outweigh any benefits. Research revealed that the presence of anti-patterns leads to the demeaning of the quality and design of the software systems, which makes the process of detecting anti-patterns in web services very crucial. In this work, we empirically investigate the effectiveness of three feature sampling techniques, five data sampling techniques, and six classification algorithms in the detection of web service anti-patterns. Experiment results revealed that the model developed by considering metrics selected by Principal Component Analysis (PCA) as the input obtained better performance compared to the model developed by other metrics. Experimental results also showed that the neural network model developed with two hidden layers has outperformed all the other models developed with varying number of hidden layers.

Keywords: Machine learning · SOA · Anti-patterns · Web services

1 Introduction:

Service Oriented Architecture (SOA) is developing as the prime integration and architectural framework in the present mind-boggling and heterogeneous computing environment. Web services are the favored standard-based approach to acknowledge SOA. Like Object-oriented programming, Web services also suffer from the deterioration of the design and Quality of Software (QoS), which leads to a poor solution called Anti-patterns. An anti-pattern is a repeated application of code or design that leads to a bad outcome. Research revealed that the presence of anti-patterns hinders the progress and maintenance of the software system. Several web service anti-patterns are discovered over time [1] and in this paper, we considered the following four anti-patterns: **GOWS**: God Object Web Service (AP1), **FGWS**: Fine-Grained Web Service (AP2), **CWS**: Chatty Web Service (AP3), and **DWS**: Data Web Service (AP4).

The primary motivation of this study is to prove that the source code metrics teamed up with the machine learning framework plays a pivotal role in the detection of web service anti-patterns. The secondary objective of this research is to develop models for the automatic detection of web service anti-patterns with the best predictive capability. Area Under a ROC Curve (AUC) and hypothesis testing approach are used to examine the relative execution of the different variations of data sampling technique: SMOTE, different feature selection techniques, and machine learning algorithms in the detection of SOA anti-patterns. In this work, we attempt to answer the following research questions:

RQ1: What is the impact of the application of data sampling techniques for developing anti-pattern prediction models?

RQ2: Is there any critical distinction between the exhibition of the models produced using a subset of features selected by applying various feature selection techniques?

RQ3: Does there exist a neural network model that outperforms all others?

2 Related Work:

Palma et al. [2] proposed a framework SODA-W for specifying and detecting the 10 anti-patterns present in the weather and finance-related data. This proposed framework achieved an accuracy of 75% and a recall of 100%. Ouni et al. [3] has used cooperative parallel evolutionary algorithms (P-EA), an automated approach to detect the anti-patterns. The idea behind their innovation is the combination of several detection algorithms executing in parallel optimization processes would give better results. Settas et al. [4] used the Protege platform, a web-based environment, to facilitate collaborative ontology editing. The model rectifies the false and imprecise information in SPARSE (using anti-pattern ontology as the knowledge base), an intelligent system that can detect anti-patterns existing in a software project. The statistical results confirm that the proposed technique outperforms other existing techniques.

3 Experimental Dataset

The data set with 226 publicly available web services that are shared by Ouni et al. on GitHub¹ are used for experiments in this paper. The dataset is of high quality, as Ouni et al. [5] who shared the dataset publicly in GitHub has manually validated the anti-patterns. The raw data that is available in the dataset is of WSDL format. A close observation of the dataset revealed that the percentage of anti-patterns present in the is varying from 5.75% to 10.62% i.e., GOWS exists in 21 out of 226 WSDL files. Similarly, FGWS, DWS, and CWS anti-patterns are present in 13, 14, and 21 out of 226 anti-patterns.

¹ <https://github.com/ouniali/WSantipatterns>.

4 Research Framework

Figure 1 illustrates the methodology for the anti-patterns prediction in web services. As discussed in Sect. 3, the dataset has a collection of web services from various domains in WSDL format. CKJM metrics are computed for each java file (A WSDL file has multiple java files) using CKJM extended tool. Further, the aggregation measures are applied on the CKJM metrics computed at the file level to obtain metrics at the system level which forms the dataset. After the formulation of the dataset, we apply different variants of SMOTE, i.e., SMOTE, BSMOTE, SVMSMOTE, SMOTEENN, and SMOTETOMEK, to address the class imbalance problem. Then we apply two different feature selection techniques, i.e., PCA and RSA, for selecting the significant features in the dataset. Further, we use the subset of features selected using PCA, RSA along with the essential metrics (SM) selected in our previous paper [6] to generate the models for the prediction of web service anti-patterns. In this paper, we use different variants of the neural network along with the ensemble technique to generate the models. Lastly, the execution of the models is evaluated using different evaluation metrics and the impact of various techniques used for generating models are speculated based on the results of hypothesis testing.

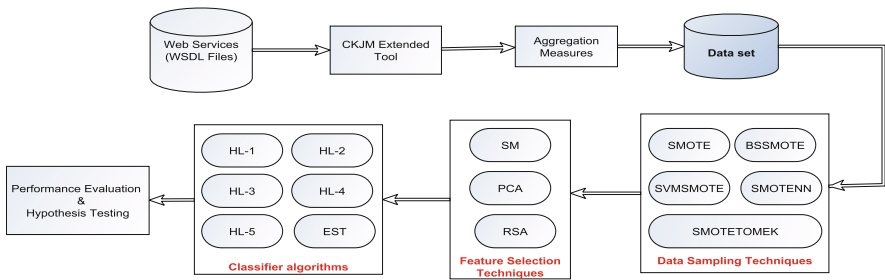


Fig. 1. Research framework for web service anti-pattern prediction

5 Experimental Results

Artificial Neural Networks (ANN) are known for their ability to learn and model non-linear and complex relationships. A neural network is a collection of interconnected nodes. The input patterns are collected by the unit nodes in the input layer and are mapped to the target variables in the output layer. In this work, we apply five different variants of neural network (NN) by changing the number of hidden k hidden layers (HL), i.e., NN with 1 HL (HL-1), NN with 2 HL (HL-2), NN with 3 HL (HL-3), NN with 4 HL (HL-4) and NN with 5 HL (HL-5). The feature matrix sets selected from different feature selection techniques are taken as input to each of the models. In addition to these models, we are also using an ensemble technique for the prediction of anti-patterns. The output of the previous models are given as input to the ensemble technique.

A five-fold cross-validation technique is applied to validate the results of the generated models. The models were trained using the original (imbalanced) dataset as well as the balanced dataset obtained after applying the data sampling techniques. Table 1 depicts the results of the models generated for the prediction of GOWS anti-pattern using five-fold cross-validation. The information present in Table 1 shows that the model developed using the neural network with 2 or 3 hidden layers have a good predictive aptness as compared to others. Similarly, the models trained using balanced data have good potential in predicting anti-patterns as compared to models generated using the original data.

Table 1. Accuracy & AUC values for GOWS anti-pattern

Feature Selection	Accuracy						AUC					
	ORG											
	HL-1	HL-2	HL-3	HL-4	HL-5	EST	HL-1	HL-2	HL-3	HL-4	HL-5	EST
SM	50.44	82.74	89.38	78.32	77.88	90.71	0.47	0.61	0.59	0.65	0.74	0.59
PCA	92.04	90.71	91.59	92.04	91.15	90.71	0.86	0.81	0.77	0.80	0.76	0.75
RSA	91.15	92.04	91.15	91.15	90.71	90.71	0.82	0.85	0.86	0.87	0.88	0.72
	SMOTE											
SM	55.61	62.93	53.17	36.59	71.22	58.54	0.56	0.63	0.53	0.37	0.84	0.47
PCA	90.73	92.68	93.41	92.68	92.20	91.95	0.96	0.97	0.96	0.96	0.95	0.93
RSA	81.95	85.12	79.02	84.39	81.46	64.88	0.90	0.88	0.85	0.87	0.86	0.71
	BSMOTE											
SM	44.63	48.29	88.05	67.80	47.07	79.51	0.45	0.49	0.88	0.68	0.36	0.87
PCA	92.93	92.93	93.41	93.41	92.68	81.95	0.96	0.97	0.97	0.96	0.96	0.91
RSA	84.39	83.17	87.32	84.88	85.85	39.51	0.89	0.87	0.89	0.89	0.88	0.59
	SVMSMOTE											
SM	40.49	69.27	59.27	82.68	80.49	65.85	0.40	0.65	0.53	0.82	0.94	0.77
PCA	91.71	93.17	93.17	92.44	94.15	76.59	0.97	0.98	0.97	0.97	0.97	0.92
RSA	90.98	90.98	92.93	91.95	93.41	75.37	0.94	0.94	0.96	0.95	0.95	0.91
	SMOTEENN											
SM	86.35	89.16	78.31	83.53	91.16	28.11	0.87	0.89	0.79	0.84	0.91	0.49
PCA	88.58	91.64	93.04	92.76	80.50	86.63	0.96	0.98	0.96	0.97	0.93	0.86
RSA	96.81	96.45	95.04	97.52	95.39	62.77	1.00	1.00	1.00	1.00	0.99	0.84
	SMOTETOMEK											
SM	46.72	53.28	39.62	71.31	74.04	67.49	0.47	0.53	0.40	0.66	0.71	0.54
PCA	90.49	90.24	91.46	91.95	93.17	74.63	0.95	0.96	0.94	0.94	0.96	0.91
RSA	85.57	89.18	87.11	71.91	90.46	65.21	0.92	0.92	0.90	0.87	0.92	0.83

6 Comparative Analysis

In this section, we have discussed and analyzed the results obtained by applying various data sampling, feature selection, and machine learning classifiers on the considered dataset for the anti-patterns prediction. The empirical analysis of the results is carried out methodically by answering the research questions defined in Sect. 1.

RQ1: What is the impact of the application of data sampling techniques for developing anti-pattern prediction models?

The impact of data sampling techniques is evaluated by analyzing the performance measures (AUC, Accuracy, and F-Measure) of anti-pattern prediction models developed before and after the application of data sampling techniques. In this paper, we employed Box-plots, and Statistical hypothesis testing to evaluate the significance and reliability of the models generated.

Comparison of the data sampling techniques based on Descriptive Statistics and Box-plots: Figure 2 depicts the box-plots for the data sampling techniques and the original data. These are useful for comparing the minimum, maximum, median, and inter-quartile range (Q1; Q3) of the various developed models. Figure 2 shows that the mean value of the model developed using the sampling technique **SMOTENN** is higher than the corresponding values of the other models. From Fig. 2, it is observed that the inter-quartile range for model generated using **SMOTE** is comparatively taller when compared to the models generated using other sampling techniques which indicates that the performance parameters computed from multiple executions are showing more variations.

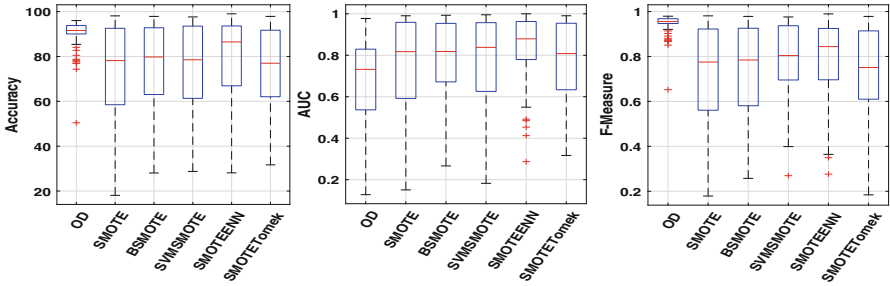


Fig. 2. Box-plot for accuracy and AUC: data sampling techniques

Comparison of the data sampling techniques based on Statistical hypothesis: We used Wilcoxon signed-rank test to evaluate the performance of the data sampling techniques statistically. The null hypothesis investigated by the Wilcoxon signed-rank test is defined as below:

Null-Hypothesis: The AUC performance value of the models commenced for web service anti-pattern prediction using various data sampling techniques is not significantly different.

The null hypothesis is accepted if the pair-wise value is greater than the considered threshold value of 0.05. From Table 2, it is noticed that most of the comparison points are having P-value, which is higher than 0.05. Hence we conclude that the null hypothesis is accepted, which means that there is no significant difference between the execution of the models generated employing various sampling techniques.

RQ2: Is there any critical distinction between the exhibition of the models produced using a subset of features selected by applying various feature selection techniques?

Table 2. Wilcoxon signed test: Data sampling techniques

	ORG	SMOTE	BSMOTE	SVMSMOTE	SMOTENN	SMOTETOMEK
ORG	1.00	0.01	0.00	0.00	0.00	0.01
SMOTE	0.01	1.00	0.65	0.57	0.05	0.99
BSMOTE	0.00	0.65	1.00	0.81	0.08	0.64
SVMSMOTE	0.00	0.57	0.81	1.00	0.20	0.51
SMOTEENN	0.00	0.05	0.08	0.20	1.00	0.03
SMOTETOMEK	0.01	0.99	0.64	0.51	0.03	1.00

The impact of the models developed by using the subset of features selected by applying various feature selection techniques is evaluated by analyzing the performance measures (AUC, Accuracy, and F-measure) on the considered dataset.

Comparison of the feature selection techniques based on Descriptive Statistics and Box-plots: Figure 3 shows the box-plots for the models trained using selected features and all features. Figure 3 show that the mean value of the model developed using the subset of features selected applying **PCA** is higher when compared to the models developed using the subset of features selected by applying other feature selection techniques.

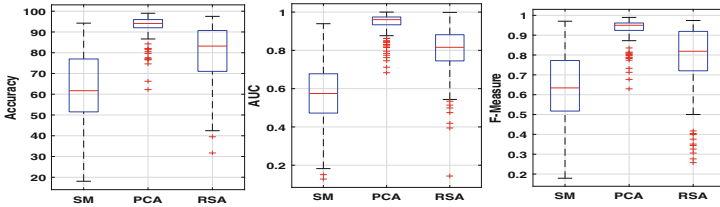


Fig. 3. Box-plot for accuracy and AUC: Feature selection techniques

Comparison of the feature selection techniques based on Statistical hypothesis: In this paper, Wilcoxon signed-rank test is used for evaluating the performance of the models generated using various feature selection techniques. From Table 3, it is observed that many of the comparison points are having P-value which is less than 0.05. Hence we conclude that the null-hypothesis is rejected, which means that, there is a noticeable difference between the performance of the models generated using the subset of features selected by applying various feature selection techniques.

RQ3: Does there exists a neural network model that outperforms all others?

The impact of the models generated using the neural network with different number of hidden layers and ensemble technique is evaluated by analyzing the performance measures (Accuracy, AUC, and F-measure) on the considered dataset.

Table 3. Wilcoxon signed test: Feature selection techniques

	SM	PCA	RSA
SM	1	4.73E-46	8.98E-26
PCA	4.73E-46	1	9.31E-28
RSA	8.98E-26	9.31E-28	1

Comparison of the classifier techniques based on Descriptive Statistics and Box-plots: Figure 4 shows the box-plots for the models generated using a neural network with a varying number of hidden layers and ensemble technique. Figure 4 show that the mean value of the model developed using the neural network with two hidden layers (HL-2) is showing preferable execution when compared to the models developed using a neural network with other numbers of hidden layers. HL-2 is showing greater performance when compared to the model generated using the ensemble technique (EST).

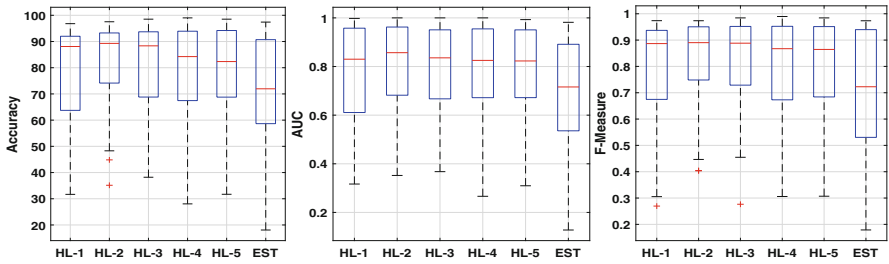


Fig. 4. Box-plot for accuracy and AUC: Classifier techniques

Comparison of the classifier techniques based on Statistical hypothesis: In this paper, Wilcoxon signed-rank test is used to evaluate the performance of the models developed using neural network with varying number of hidden layers and the ensemble technique. Considering only the models developed using the neural network with varying number of hidden layers, we notice from Table 4, that many of the comparison points are having P-value, which is higher than 0.05, from which we conclude that the null-hypothesis is accepted, which means that, there is no significant variation between the performance of the models generated using a neural network with varying number of hidden layers. If we consider the model developed by applying neural networks with any number of hidden layers and ensemble technique, it is observed that most of the comparison points are having a p-value of less than 0.05. Hence we conclude that the null-hypothesis is rejected and infer that there is a significant variation between the execution of the models generated using neural network (HL-1, HL-2, HL-3, HL-4 and HL-5) and the ensemble technique (EST).

Table 4. Wilcoxon signed test: Classifier techniques

	HL-1	HL-2	HL-3	HL-4	HL-5	EST
HL-1	1.00	0.53	1.00	0.92	0.95	0.00
HL-2	0.53	1.00	0.62	0.62	0.58	0.00
HL-3	1.00	0.62	1.00	0.99	0.97	0.00
HL-4	0.92	0.62	0.99	1.00	0.84	0.00
HL-5	0.95	0.58	0.97	0.84	1.00	0.00
EST	0.00	0.00	0.00	0.00	0.00	1.00

7 Conclusion

The principle inference of this work is that neural network with less number of hidden layers can be used for the effective prediction of web service anti-patterns. In this paper, the application of five data sampling techniques along with the original data, three feature selection techniques and six machine learning classifier techniques, i.e., neural network (HL-1, HL-2, HL-3, HL-4 and HL-5) and the ensemble technique (EST) is investigated empirically. The significant finding of this experimental work is that feature selection techniques play a crucial role in removing irrelevant features. Experimental results reveal that SMOTEENN is showing better performance. We also infer that the model developed by considering metrics selected by Principal Component Analysis (PCA) as the input obtained better performance when compared to the model developed by other metrics. Experimental results also show that the neural network model developed with two hidden layers has outperformed all the other models developed with varying number of hidden layers.

References

1. Dudney, B., Asbury, S., Krozak, J.K., Wittkopf, K.: J2EE Antipatterns. John Wiley & Sons, Hoboken (2003)
2. Palma, F., Moha, N., Tremblay, G., Guéhéneuc, Y.-G.: Specification and detection of SOA antipatterns in web services. In: Avgeriou, P., Zdun, U. (eds.) ECSCA 2014. LNCS, vol. 8627, pp. 58–73. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09970-5_6
3. Ouni, A., Kessentini, M., Inoue, K., Cinnéide, M.O.: Search-based web service antipatterns detection. *IEEE Trans. Serv. Comput.* **10**(4), 603–617 (2017)
4. Settas, D., Meditskos, G., Bassiliades, N., Stamelos, I.G.: Detecting antipatterns using a web-based collaborative antipattern ontology knowledge base. In: International Conference on Advanced Information Systems Engineering, pp. 478–488. Springer (2011). https://doi.org/10.1007/978-3-642-22056-2_50

5. Ouni, A., Kessentini, M., Inoue, K., Cinnéide, M.O.: Search-based web service antipatterns detection. *IEEE Trans. Serv. Comput.* **10**(4), 603–617 (2015)
6. Tummalapalli, S., Kumar, L., Neti, L.B.M.: An empirical framework for web service anti-pattern prediction using machine learning techniques. In: 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), pp. 137–143. IEEE (2019)