INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

# Impact of Unbalanced Classification on the Performance of Software Defect Prediction Models

**K J Eldho**[1]*

**1** Assistant Professor, Department of Computer Science, Mary Matha Govt Aided Arts & Science College, Mananthavady, Kerala

OPEN ACCESS

**Received:** 25.11.2021

**Accepted:** 23.01.2022

**Published:** 16.02.2022

**Citation:** Eldho KJ (2022) Impact of Unbalanced Classification on the Performance of Software Defect Prediction Models. Indian Journal of Science and Technology 15(6): 237-242. https://doi.org/10.17485/IJST/v15i6.2193

* **Corresponding author**.

eldhorvs@gmail.com

**Funding:** None

**Competing Interests:** None



Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

## Abstract

**Objectives:** To propose a suitable imbalanced data classification model to split the dataset into two new datasets and to test the created imbalanced dataset by the prediction models. **Methods:** The imbalance defect data sets are taken from the PROMISE library and used for the performance evaluation. The results clearly demonstrate that the performance of three existing prediction classifier models, K-Nearest Neighbor (KNN), Naive Bayes (NB), and Back Propagation (BPN), is very susceptible in terms of unbalance of classification, while Support Vector Machine (SVM) and Extreme Learning Machine (ELM) are more stable. **Findings:** The outcome of this research reveals that applied SVM and ELM machine learning models improves the performance in defect prediction and records 29% more than KNN, and 19% more than NB and BPN. **Novelty:** According to the findings of a comprehensive study, the proposed machine learning new classification imbalance impact analysis method outperforms the existing ones in order to transform the original imbalance data set into a new data set with an increasing imbalance rate and be able to select models to evaluate different predictions on the new data set.

**Keywords:** Software Fault Prediction Model; Imbalance Problem Classification; Artificial Intelligence; Smart Debugging; Unbalanced Classification

## 1 Introduction

Defect prediction is essential in the software field in terms of quality and reliability, and it is one of the major comparative research areas in the modern software engineering approach. Numerous defect prediction models have been introduced for the class imbalance problem by means of the continuous development of machine learning and data mining. In machine learning, classifiers are created to eliminate errors and increase accuracy. Classification imbalance has gradually become the current dominant research hotspot in software engineering. Generally, unbalanced classification refers to the phenomenon that the sample size distribution among different categories is unbalanced. For example, in the binary classification problem, when the sample size of the two categories differs greatly, the classification imbalance problem appears. In real time, classification imbalance problems are common and need to be addressed to

maintain quality. The imbalance issues such as text classification, web fault prediction [1], credit card fraud detection [2], High error rate of classification models etc [3]. However, when dealing with the imbalanced data by focusing on prediction model the traditional classification method becomes inefficient in terms of error rate and accuracy varying for the datasets [4]. Traditional defect prediction techniques mainly target coarse-grained software entities, such as files, modules, or packages for prediction are discussed and in-depth analysis for prediction models portrayed in cloud environment using deep learning method which have few drawbacks in transforming the unbalanced dataset [5]. SLD approach is proposed by authors using deep learning on static codes but fails in classification of unbalanced formulated data [6]. K-Nearest Neighbor (KNN), Naive Bayes (NB), Back Propagation (BPN) models are used and compared for defect prediction to find the same defects with different classifiers [7]. Classifiers Compositional C1 level defect prediction is presented where the error rate is high which leads to minimal accuracy [8], CP (Cross Project) defect prediction software model was introduced where it has drawbacks on TP and TN rate analysis [9], Heterogeneous model and multi-source hinders performance models were proposed for prediction and all these approaches has drawbacks in transformation of new dataset from old after classifying the imbalanced data and shows minimal accuracy [10]. Enhanced CCFDB defect prediction model was developed using SVM for parameter optimization and discussed many methods to solve the problem of classification imbalance such as sampling methods, including over-sampling and under-sampling [11].

Machine Learning prediction model was utilized for medical datasets for acute organ failure in critical patients [12]. Traditional classification model believes that the misclassification and the SDP model was presented for noise and class imbalance defects to remove the noisy datas which have drawbacks on high error rate in transformation of unbalanced dataset [13]. Classifier ensemble method for high dimensional data classification was portrayed to overcome the baseline models which have minimal drawbacks on predicting the imbalanced datas in an optimized manner [14]. Some models improve classification performance by gathering the prediction results of multiple data sets. In general, the performance of the integrated model is better than that of a single model. Although ensemble learning is not proposed to solve the problem of unbalanced classification, it can achieve better results when dealing with the problem of unbalanced classification. Hybrid prediction model (i.e., classification models) using DBSCAN and RF, SMOTE was derived and during the analysis it was found that the FP and FN rate varies on datasets during different iteration [15]. How to choose a reasonable prediction model? In addition, the prediction model itself may also be affected by the imbalance of classification. Which prediction models have more stable performance? If we can grasp the performance stability of different prediction models when the classification is unbalanced, then we can choose a reasonable prediction model in a practical application to better guide the software defect prediction work. The classifier chains for class imbalance via RS (Random Sampling) were discussed and the analysis was conducted, and it was found that there was an issue with imbalanced data transformation to a new dataset [16]. The CSALB approach is utilised for imbalanced fault diagnosis by the authors, and here the defect prediction is not efficient in terms of the transformation of the dataset [17]. QoS-IWDARP is also used to find accuracy in network data sets for early route prediction, but it also has a few drawbacks in the removal of noisy data [18]. The authors identified a cost-effective method and created an application for NASA software defects; this approach has few drawbacks in the knowledge discovery process [19]. To improve software testing effectiveness, an AI-based SDP named Defect Prediction as a Service was created, along with six best defect prediction models [20]. Eleven software defect prediction models are identified and compared to improvise the quality and security in software testing which provides accuracy of 80% [21].

The main contribution of this article is to i) introduce a classification imbalance impact analysis method and transform the original imbalance data set into a new data set with an increasing imbalance rate, and select different prediction models to predict the new data set to evaluate different predictions. The degree of stability of the prediction model is analyzed when the classification is unbalanced. ii) Experiments were carried out on 3 existing typical prediction models (KNN, NB, and BPM) and 2 proposed machine learning and extreme learning methods (SVM and ELM) to show the stability in classification of unbalanced data. iii) The performance stability of different existing prediction models (KNN, NB, BPM) is derived in a PROMISE data library when the classification is unbalanced to showcase the proposed SVM and ELM as reasonable prediction models for practical applications, which has a certain guiding role for the research of software defect prediction.

The proposed method constructs a new data set with an increasing imbalance rate from the original unbalanced data set and then selects eight typical classification models as defect prediction models, respectively, for the constructed new data set and uses ROC (Receiver Operating Characteristic Curve (Area under the Curve). The index is used to evaluate the classification performance of different prediction models, and at the same time, different coefficients of variation are used to predict the different coefficients of the prediction model. The experimental results show that the performance of the three existing prediction models, BPN, NB, and KNN, decreases with the increase of the imbalance rate, indicating that the performance of these three models is very susceptible to classification imbalance, and that SVM and ELM outperform with high performance in defect prediction.

## 2 Proposed Methodology

In order to make the improvised prediction model, it is necessary to clean the noisy or imbalanced data in the repository. The quality of datasets is highly essential for experimentation and evaluation. Machine learning methods are highly dominant in dealing with imbalanced datasets, minimizing error rates and removing noisy datasets from large numbers of datasets. The proposed SVM and ELM methods aim to evaluate the datasets in the PROMISE library to transform the original imbalanced datasets into the new ones for imbalanced classification. The preliminary statistical results are shown in Table 1, including the name (abbreviation) of the prediction model, the method employed, and quotations. KNN, NB, BPN, SVM and ELM defect prediction models are employed to carry out the experiment.

### 2.1 Impact of Unbalanced Classification on the Performance of Software Defect Prediction Models

The number of non-defective samples in the data set is much higher than the number of defective samples, a phenomenon called unbalanced classification. The data set D = {x1, x2,..., xn}, xi∈ Rd (i = 1, 2, ..., n), which includes a number of samples, and each sample contains a In addition, it also includes a category feature to mark the category of the sample, i.e., defective or non-defective. According to the characteristics of the category, the data set can be divided into defective type C1 and non-defective type C2. The number of samples is n1 and n2, respectively, and n = n1 + n2. As a result, the imbalance rate of the data set D (Imbalance Ratio) [12] is defined as the ratio of the number of non-defective samples, n2, to the number of defective samples, n1, which is reduced to IR=(n2/n1). In general, n2 > n1, i.e., IR > 1. The greater the IR value, the greater the degree of imbalance in the data set classification, and vice versa.

### 2.2 SVM – Support Vector Machine for unbalanced datasets

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, each data item is plotted as a point in an n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate to derive the unbalanced data in an optimized manner.

### 2.3 ELM – Extreme Learning Machine for transformation

In ELM, the number of defects is identified and TP and TN are performed. The transformation of the original imbalanced dataset is classified into new datasets and used ROC for migration, where the parameters of hidden or noisy are removed. The imbalanced rate is calculated. The number of defective samples (n1) and the number of non-defective samples (n2) with size is calculated. The tasks of classification and transformation are accomplished by running extensive tests on imbalanced classification datasets with various class ratios in the PROMISE data library. A set of data sets with different imbalance rates is needed in order to explore the impact of classification imbalance on the performance of prediction models. That is, the changes in the performance of each prediction model in the case of classification imbalance. Therefore, this paper designs a new data set construction algorithm to transform the original unbalanced data set into a new data set with successively increasing unbalance rates. The specific process is shown in Algorithm 1.

**New Data Set Construction Algorithm**

Input: DataSet - The original unbalanced data set
Output: NewDataSet - New data set
1. According to category and characteristics DataSet is divided into DefectSet and NonDefectSet;
2. Number of defective samples n1=DefectSet.Size();
3. Number of non-defective samples n2=Non NonDefectSet.Size();
4. Imbalance rate r=Math.floor(n2/n1);
5. Form newDataSet=DefectSet;
6. Form restNonDefectSet=NonDefectSet;
7. WHILE restNonDefectSet=NonDefectSet;
8. Randomize the restNonDefectSet
9. IF restNonDefectSet.Size() >=2n1
10. choose n1 samples randomly from restNonDefectSet
Save as newDataSet

11. Remove the selected sample from the restNonDefectSet;
12. ELSE
13. save the remaining sample in restNonDefectSet as new DataSet;
14. restNonDefectSet = null;
15. END IF
16. Save newDataSet;
17. END WHILE
18. RETURN all new datasets as newDataSet;

## 3 Results and Discussions

The performance analysis was conducted against KNN, NB, and BPN [7] with SVM and ELM by using PROMISE datasets in the Weka tool under the Windows platform to evaluate the performance stability of different prediction models. Software defect prediction is a two-class classification problem, and its prediction process will produce 2 different results, as shown in Table 2, where defects are positive cases, and no defects are negative cases. The row represents the actual category, and the column represents the predicted category. In the forecasting process, it is first necessary to select reasonable indicators to evaluate the performance of each forecasting model. The area under the ROC curve is used for evaluation.

### 3.1 Test Dataset

The proposed method selects 2 unbalanced classification data sets. Basic information is shown in Table 1. These data sets are all defect data sets in the PROMISE library. The first column indicates the name of the dataset, the second column indicates the development language of the dataset program; the third column indicates the number of features contained in the dataset, that is, the feature dimension; the fourth column indicates the samples in the dataset total, which describes the size of the data set, including small-scale (a few hundred), medium-scale (thousands), and large-scale (tens of thousands); the fifth to seventh columns represent the number of defective samples and non-defective samples in the data set, respectively. The number of samples and the defect rate; the eighth column represents the unbalance rate of the data set, which is calculated as IR=($n_2/n_1$). That is, the ratio of the number of samples with no defects to the number of samples with defects is an integer. The larger the value, the more unbalanced the classification of the data set. Jedit-4.3 and Tomcat are open source data sets, and the features they contain are class-level CK metrics, which comprehensively consider the inheritance, coupling, and cohesion in object-oriented programs and also measured the correlation between software features and defects.

**Table 1. Experimental Dataset**

| Dataset Name | Lan-guage | Number of Features | Number of Samples | Number of Defective Samples | Number of non-Defective Samples | Defect Rate (%) | Imbalance Rate |
|---|---|---|---|---|---|---|---|
| Jedit-4.3 | Java | 20 | 492 | 11 | 481 | 2.24 | 43 |
| Tomcat | Java | 20 | 858 | 77 | 781 | 8.97 | 10 |

The prediction result is jointly determined by the data set and the prediction model. For a certain data set, which includes many software features, all these features are used to train the prediction model when feature selection is not performed. All the features in the above data set are used to train the new machine learning prediction model. In order to explore the difference in performance stability of different prediction models, KNN, NB, and BPN with SVM and ELM are compared in performance analysis [7].

### 3.2 Comparison of Performance Stability of Different Prediction Models

A binary classification problem, such as fault-prone (positive) and not fault-prone (negative), has four possible prediction outcomes: True Positive (TP) (i.e., correctly classified positive instance), False Positive (FP) (i.e., negative instance classified as positive), True Negative (TN) (i.e., correctly classified negative instance), and False Negative (FN) (i.e., positive instance classified as negative). The four values form the basis for several other performance measures that are well known and commonly used for classifier evaluation. Overall Accuracy (OA) provides a single value that ranges from 0 to 1. It can be calculated by the equation, OA = (|T P| + |T N|)/N, where N represents the total number of instances in a dataset. While overall accuracy facilitates model performance comparisons, it is not always regarded as a reliable performance metric, particularly in the presence of class imbalance. The area under the ROC (receiver operating characteristic) curve is a single-value measurement that originated in

the field of signal detection. The value of the AUC ranges from 0 to 1. The ROC curve describes the trade-off between True Positive Rate (TPR) TPR = TP/(TP+FN) and False Positive Rate (FPR) FPR = FP/(FP+TN). A classifier that provides a large area under the curve is preferable to a classifier with a smaller area under the curve. The TPR (True Positive Rate) refers to the ratio of the number of correctly predicted positive cases to the actual number of positive cases, that is, the ratio of the number of correctly predicted defective samples to the actual number of defective samples. The FPR (False Positive Rate) refers to the ratio of the number of false positive cases to the actual number of negative cases; that is, the ratio of the number of falsely predicted defective samples to the actual number of non-defective samples.

For a specific prediction model and training data set, the prediction result corresponds to a point on the ROC curve. By adjusting the threshold of the model, a curve passing through (0, 0) and (1, 1) can be obtained below the curve. The area of A is the value of A. In particular, the value range of AT is 0 to 1. When AT is 0.5, it represents the performance of the random guessing model. The larger the value of A, the better the performance of the model. Therefore, a good prediction model should be as close as possible to the upper left corner of the coordinate system. Use the prediction models presented in [7], namely KNN, NB, BPN, and the data set Jedit and Tomcat, to conduct combined experiments. First, select a data set, and use Algorithm 1 to transform the data set into a new data set with an increasing imbalance rate (ie, IR = 1, 2, ..., r); then, use the prediction models to predict the new data set separately to obtain a set of AT values under different imbalance rates, which are recorded as the set S={AUC1, AUC2,..., AUCr; Finally, through the Coefficient of Variation (CV) of this group of AT values, to evaluate the performance stability of different prediction models under different imbalance rates.

hows the experimental results of each prediction model on different data sets, including the mean $\mu$, standard deviation $\sigma$, and coefficient of variation CV. The larger the coefficient of variation (CV) shows, the more unstable performance of the prediction model, which has a greater impact of imbalance in classification on the performance of the prediction model.

**Table 2. Evaluation Results**

| Prediction Models | Jedit | | | Tomcat | | |
|---|---|---|---|---|---|---|
| | Mean($\mu$) | Std($\sigma$) | CV | Mean($\mu$) | Std($\sigma$) | CV |
| **Existing SDP Models** | | | | | | |
| KNN [7] | 0.513 | 0.021 | 1.710 | 0.582 | 0.024 | 2.135 |
| NB [7] | 0.524 | 0.021 | 1.902 | 0.621 | 0.026 | 2.102 |
| BPN [7] | 0.581 | 0.022 | 1.903 | 0.624 | 0.053 | 2.821 |
| **Proposed SDP Models** | | | | | | |
| **SVM** | **0.612** | **0.012** | **2.102** | **0.734** | **0.021** | **6.031** |
| **ELM** | **0.721** | **0.020** | **2.204** | **0.768** | **0.032** | **7.875** |

It can be seen from Table 2 that the SVM, and ELM three prediction models have relatively high CV values on both datasets, indicating that the performance of these proposed models is highly stable to the imbalance of classification whereas the performance of models such as KNN, NB, and BPN [7] is low compares to SVM and ELM. In addition, the sample distribution differences between different data sets will also affect the performance of the prediction model to a certain extent. Therefore, the performance of the same model on individual data sets may be different from the performance on other data sets. For example, the KNN model only shows a slight instability on the Jedit data set (the CV value is 1.71%), while the BPN model only shows obvious instability on the Jedit data set (the value of CV is 1.903%), because the number of defective samples in this data set is too small. Whereas the KNN, NB, and BPN [7] model performs little better in Tomcat dataset. This makes the initially constructed new data set too small, which affects the performance stability of the prediction model. SVM and ELM shows the stability in SDP for both Jedit and Tomcat datasets.

Finally, there are external factors, such as the quality of the data set, which may affect the evaluation of the stability of the predictive model's performance by the methods used in this paper. Therefore, more sufficient experiments on data sets of different scales, different defect rates, and different imbalance rates ensure that all experimental data sets are real data sets in defect prediction and are also the most commonly used defect data sets to ensure the validity and reliability of the prediction results.

## 4 Conclusion

The research study proposed a new model with machine learning techniques such as SVM and ELM to classify the imbalanced data in the PROMISE library to evaluate the software defect prediction. This method transforms the original unbalanced data set into a new set of data sets with an increasing unbalance rate, and then selects a typical prediction model to predict and

evaluate the new data set, respectively. The experimental results show that new defect prediction machine learning techniques SVM and ELM outperform KNN, NB, and BPN in terms of imbalance rate and data classification. SVM records 29% more than KNN up to 2.102 and ELM records 19% more than NB and BPN up to 2.204 towards Jedit and 6.031 and 7.035 towards Tomcat. The limitations of the study lie in the fact that the prediction and classification accuracy level might vary depending on the datasets. In the future, the model can be improvised to predict the faulty data in large datasets in an optimized manner as software quality and reliability are the main concerns in the modern software engineering era.

## References


1) Pandey SK, Tripathi AK. Class Imbalance Issue in Software Defect Prediction Models by various Machine Learning Techniques: An Empirical Study. *2021 8th International Conference on Smart Computing and Communications (ICSCC)*. 2021;2021:58–63. Available from: https://doi.org/10.1109/ICSCC51209.2021.9528170.
2) Makki S, Assaghir Z, Taher Y, Haque R, Hacid MS, Zeineddine H. An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection. *IEEE Access*. 2019;7:93010–93022. Available from: https://dx.doi.org/10.1109/access.2019.2927266.
3) Thabtah F, Hammoud S, Kamalov F, Gonsalves A. Data imbalance in classification: Experimental evaluation. *Information Sciences*. 2020;513:429–441. Available from: https://dx.doi.org/10.1016/j.ins.2019.11.004.
4) Mohammed A, Podila PSB, Davis RL, Ataga KI, Hankins JS, Kamaleswaran R. Machine learning predicts early-onset acute organ failure in critically ill patients with sickle cell disease. *bioRxiv*. 2019;p. 614941–614941. Available from: https://doi.org/10.1101/614941.
5) Liu W, Wang B, Wang W. Deep Learning Software Defect Prediction Methods for Cloud Environments Research. *Scientific Programming*. 2021;2323100:1–11. Available from: https://doi.org/10.1155/2021/2323100.
6) Majd A, Vahidi-Asl M, Khalilian A, Poorsarvi-Tehrani P, Haghighi H. SLDeep: Statement-level software defect prediction using deep-learning model on static code features. *Expert Systems with Applications*. 2020;147:113156–113156. Available from: https://dx.doi.org/10.1016/j.eswa.2019.113156.
7) Bowes D, Hall T, Petrić J. Software defect prediction: do different classifiers find the same defects? *Software Quality Journal*. 2018;26(2):525–552. Available from: https://dx.doi.org/10.1007/s11219-016-9353-3.
8) Xia X, Lo D, Pan SJ, Nagappan N, Wang X. HYDRA: Massively Compositional Model for Cross-Project Defect Prediction. *IEEE Transactions on Software Engineering*. 2016;42(10):977–998. Available from: https://dx.doi.org/10.1109/tse.2016.2543218.
9) Wu J, Wu Y, Niu N, Zhou M. MHCPDP: multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder. *Software Quality Journal*. 2021;p. 1–26. Available from: https://dx.doi.org/10.1007/s11219-021-09553-2.
10) Shrikanth NC, Majumder S, Menzies T. Early Life Cycle Software Defect Prediction. Why? How. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 2021;p. 448–459. Available from: https://doi.org/10.1109/ICSE43902.2021.00050.
11) Rtayli N, Enneya N. Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *Journal of Information Security and Applications*. 2020;55:102596–102596. Available from: https://dx.doi.org/10.1016/j.jisa.2020.102596.
12) Mohammed A, Podila PSB, Davis RL, Ataga KI, Hankins JS, Kamaleswaran RS. Machine learning predicts early-onset acute organ failure in critically ill patients with sickle cell disease. *bioRxiv*. 2019;p. 614941–614941. Available from: https://doi.org/10.1101/614941.
13) Pandey SK, Tripathi AK. An empirical study toward dealing with noise and class imbalance issues in software defect prediction. *Soft Computing*. 2021;25(21):13465–13492. Available from: https://dx.doi.org/10.1007/s00500-021-06096-3.
14) Xu Y, Yu Z, Chen CLP, Cao W. A Novel Classifier Ensemble Method Based on Subspace Enhancement for High-Dimensional Data Classification. *IEEE Transactions on Knowledge and Data Engineering*. 2021;p. 1–1. Available from: https://dx.doi.org/10.1109/tkde.2021.3087517.
15) Ijaz M, Alfian G, Syafrudin M, Rhee J. Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. *Applied Sciences*. 2018;8(8):1325–1325. Available from: https://dx.doi.org/10.3390/app8081325.
16) Liu B, Tsoumakas G. Dealing with class imbalance in classifier chains via random undersampling. *Knowledge-Based Systems*. 2020;192:105292–105292. Available from: https://dx.doi.org/10.1016/j.knosys.2019.105292.
17) Peng P, Zhang W, Zhang Y, Xu Y, Wang H, Zhang H. Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis. *Neurocomputing*. 2020;407:232–245. Available from: https://dx.doi.org/10.1016/j.neucom.2020.04.075.
18) Nithyanandh S. Quality of service enabled intelligent water drop algorithm based routing protocol for dynamic link failure detection in wireless sensor network. *Indian Journal of Science and Technology*. 2020;13(16):1641–1647. Available from: https://dx.doi.org/10.17485/ijst/v13i16.19.
19) Siers MJ, Islam MZ. Novel algorithms for cost-sensitive classification and knowledge discovery in class imbalanced datasets with an application to NASA software defects. *Information Sciences*. 2018;459:53–70. Available from: https://dx.doi.org/10.1016/j.ins.2018.05.035.
20) Pandit M, Gupta D, Anand D, Goyal N, Aljahdali HM, Mansilla AO, et al. Towards Design and Feasibility Analysis of DePaaS: AI Based Global Unified Software Defect Prediction Framework. *Applied Sciences*. 2022;12(1):493–493. Available from: https://dx.doi.org/10.3390/app12010493.
21) Pal S, Sillitti A. A Classification of Software Defect Prediction Models. *2021 International Conference "Nonlinearity, Information and Robotics" (NIR)*. 2021;2021:1–6. Available from: https://doi:10.1109/NIR52917.2021.9666110.