Code Clone Detection Using Metrics Based Technique and Classification using Neural Network

Sukhpreet Kaur [1], Prof. Manpreet Kaur[2]
[1,2]*Department of Computer Science and Engineering*
Baba Banda Singh Bahadur Engineering College
Fatehgarh Sahib and Punjab Technical University Jalandhar, Punjab – India

Abstract-Code clone are the duplicated code which degrade the software quality and hence increase the maintenance cost detection of clones in large software system is very tedious tasks but it is necessary to improve the quality of software system product. In this paper, two types of investigation are performed. The first is to apply ant colony optimization technique to generate optimized dataset and second to predict the code clones by using back propagation neural network classifier. Software metrics like Loc, total function, functions repetitive, function overloading, private variable, public variable which are generated from some .java programs using MATLAB tool. Clone detection using software metrics is the best technique of code clone detection and the back propagation neural network gives more accurate result with faster training and testing of the neural network.

Keywords- code fragments, code clone, clone prediction, software metrics, back propagation neural network, precision, recall, accuracy etc.

I. INTRODUCTION

Software code clones are the similar or identical parts of the source code which may be inserted either by mistake or knowingly. But the presence of these code clones may decrease the software quality and hence increase the maintenance cost. So the detection and removal of code clones are necessary in the software products. Copying existing code fragments and pasting them with or without modifications into other sections of code is a frequent process in software development. The copied code is called a software clone and the process is called software cloning. Prediction of code clone plays vital role to improve the quality of software product. In this research work two types of investigation are performed. The first is to apply ant colony optimization technique to generate optimized dataset and second to predict the code clones by using back propagation neural network. In this research, multiple input neurons with one or more hidden layers and single output neuron is used and multilayer perception is used to calculate the output neuron. The output neuron predicts whether the software program has the code clone or not.

Example of code clones:-

```
1.  int sum = 0 ;
2.  void foo ( I t e r a t o r i t e r ){
3.  for ( item = f i r s t ( i t e r ) ; has more ( i t e r ) ; item = next
    ( i t e r ) ) {
4.  sum = sum + value ( item ) ;
5.  }
6.  }
7.  int bar ( I t e r a t o r i t e r ){
8.  int sum = 0 ;
9.  for ( item = f i r s t ( i t e r ) ; has more ( i t e r ) ; item = next
    ( i t e r ) ) {
10. sum = sum + value ( item ) ;
11. }
12. }
```

II. RELATED RESEARCH

The literature review of code clone detection and analysis begins with basic concept of code clone detection terminology.

A. Code fragment:

A code fragment is any sequence of code line with or without comments. It can be of any granularity level for example function definition, begin –end block, or sequence of statements.

B. Types of Clones:

Type1-(exact clones): Program fragments which are identical except for variations in white space and comments.

Type2-(renamed/parameterized clones):Program fragments which are structurally/syntactically similar except for changes in identifiers, literals, types, layout and comments.

Type3-(near miss clones): Program fragments that have been copied with further modifications like statement insertions/deletions in addition to changes in identifiers, literals, types and layouts.

Type4-(semantic clones): Program fragments which are functionally similar without being textually similar.

C. Precision, recall and Accuracy:

$$\text{Precision (P)} = \frac{\text{Number of clones correctly found}}{\text{Total number of clone found}} \text{ or } (1\text{-FAR})$$

$$\text{Recall (R)} = \frac{\text{Number of clones found correct}}{\text{Total number of clone found}} \text{ or } (1\text{-FRR})$$

$$\text{Accuracy (A)} = (1\text{-fAR} + \text{fRR})*100$$

FAR- False acceptance rate, FRR- False rejection rate

Roy Ck [1] did comparison of different techniques of clone detection such as textual approach, lexical approach, semantic approach and metric based approach and also comparing and evaluating clone detection tools such as Duploc, simian and NICAD. They proposed that NICAD tool is the best among all others. Moreover they explain four category of clone viz-Type-1, Type-2, Type-3 and Type-4.

Roy CK et.al [2] also survey the state of the art in clone detection research. Firstly they describe the clone terms commonly used in the literature along with their corresponding mappings to the commonly used clone types. Secondly they give the review of existing clone detection approaches and techniques.

Gayathri et.al [3] detect the different types of clones using different algorithm like textual analysis, metric based distance

Test cases	X(min)	Y(max)	X(max)	Y(min)
Test 1	1	100	100	1.415
Test 2	1	100	100	1.631
Test 3	1	105.5	100	1.499
Test 4	1	76.06	100	0.356
Test 5	1	22.18	100	0.755
Test 6	1	100	100	1.415
Test 7	1	66.98	100	1.712

algorithm and mapping algorithm. The detected clones are-extract clone, renamed clone, gapped cloned and semantic clone used clone detection and metrics to evaluate quality. Then they discuss several approaches used in clone detection. Metric based clone detection approach uses the metric based distance algorithm. Then they compared different types of approach using different algorithm and calculate their metrics, speed, cost and quality.

Pavitdeep et.al [4] developed a tool "Software Quality assurance tool" in dot net framework using C# as programming language. This tool generates the software code metrics for C# projects using the AST technique. This tool works at method and class level metrics. This tool also detects the clones of Type-1 and Type-2.

They also compared different types of tools [5] with each other and explaining their merits and demerits of each tool. The tool of software quality assurance predicts and calculated the metrics of modern languages like c# using the technique of abstract syntax tree (AST).

Kodhai.A and Kanmani.[5] present a novel code clone detection approach using textual analysis and software metrics. 12 software metrics at method level instead of 7 are used. It has also been implemented as a tool using Java. The tool efficiently and accurately detects type-1, type2, type-3 and type-4 clones found in source codes at method level in JAVA open source code projects. The main limitation of this research is that it is language dependent and detect clone in JAVA open source project only.

Rubala et.al[6] did research of code clone detection in web based application. Web based applications used the commerce functionality in web sites. Scripting languages such as ASP, JSP, PHP etc are used in the development of web sites in which code duplication practice usually involved in making of several web pages. Hybrid approach (textual and metric based) is used. The proposed method is implemented as a tool in .NET. A set of 7 existing function level metrics are used for the detection of all types of clone functions in web application. The proposed tool gives its evaluated result in precision and recall parameter which then further compared with the other existing tool called e Metrics. The result of comparison showed that the value of precision and recall in term of percentage with the proposed tool using .NET gives higher value with accuracy than the e Metrics tool. The limitation of this research is problem with working on larger and even more complex system.

Dr. C.R.K Reddy et.al [8] also uses metrics and textual based technique to find the code clones in a software projects. They use a tool to implement the proposed work in JAVA. The technique easily deals with type-1 and type-2 clones.

Yogita Sharma et.al [9] present hybrid approach for detection of code clones. In this research, object oriented metrics and text based technique are used for the detection of exact clones. An automated tool for exact code clone detection was developed in VB.Net which calculates the metrics of the C/C++ projects and also performs the analysis of code clone detection that is which project function or the class had the code clone by using textual comparison. This approach has the limitation that it is only limited for C/C++ projects or software.

Thwin and Queh [10] present a neural network modeling technique along with regression analysis called GRNN to improve the quality of software products. In this paper, ward neural network and General regression neural network are used. First on predicting the number of defects in a class and the second on predicting the number of lines changed per class.

Sukhpreet and Satwinder [16] in this research, various type of metrics based code clone detection approach and techniques are discussed. From the discussion it is concluded that clone detection using software metrics is the best technique of code clone detection.

III. PROPOSED WORK

A. Dataset

The dataset for the analysis of metric based clone detection has been collected from any offline or online software projects which are developed in some .java programming and Software metrics like Loc, total function, repeated function, private variable, public variable which are generated from some .java programs using MATLAB tool.

B. Software metrics

Metrics are very useful in the prediction of software clones.

- LOC (line of code)-total number of lines in a program.
- Total function- total functions used in a program.
- Repeated function –how many times one function repeated in a program.
- Private variable –total private function used in a program.
- Public variable- total public function used in a program.

C. Ant Colony Optimization (ACO) technique

In this research work two types of investigation are performed. The first is to apply ant colony optimization technique to generate optimized dataset because the given dataset has more complexity so to reduce this complexity of

dataset and obtain optimized dataset for better results ACO is used and second to predict the code clones by using back propagation neural network. ACO is one of the best optimize technique.

D. Back Propagation Neural Network (BPNN): Neural network trained with back propagation learning algorithm are the most popular neural network. They are applied to variety of problems. A BPNN consists of neurons that are ordered into layers (input, hidden and output layers) as shown in fig 2. In this research, clone is chosen as the dependent variable and 6-object-oriented metrics as the independent variables. Prediction of code clone is performed with the help of multilayer perception with sigmoid activation function. The output neuron predicts whether the classes have the code clone or not.

Table 1 .Optimized reduced dataset
X - no. of iterations, Y- feature values

No. of Code Files	FAR (Detect Clone)
1	0.0006
2	0.0014
3	0.00214
4	0.00288
5	0.00363
6	0.00437
7	0.005185
8	0.00592
9	0.00666
10	0.007259

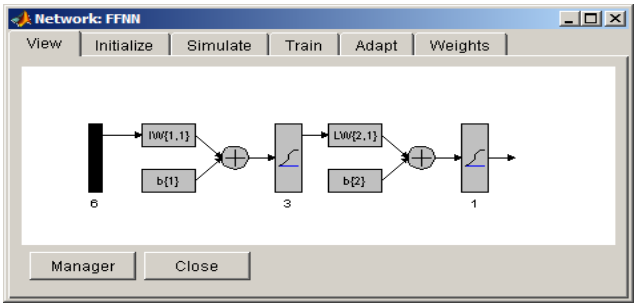
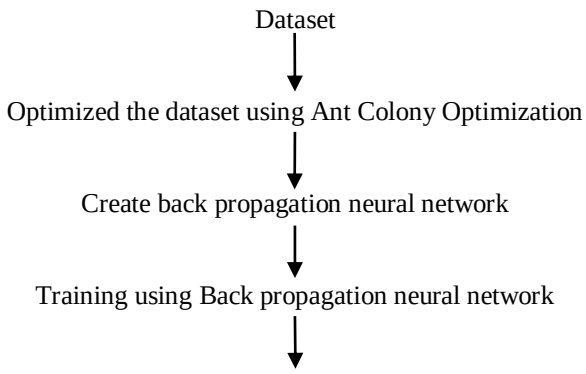


Fig. 1 Architecture of multilayer feed forward neural networK

E. Clone prediction process:



BPNN Predictions results

F. Neural Network Modeling

Neural Network Toolbox (NNTOOL) in MATLAB software provides tools for designing, implementing, visualizing, and simulating neural networks. Using this we create Multilayer perception neural networks with five input neuron and one output neuron and trained them using the dataset. Neural network are used for applications where formal analysis would be difficult or impossible.

G. Prediction of Clones

For the prediction of code clone, data is collected and optimized using ant colony optimization technique. Then a single layer perception neural network using activation function is created and trained with the given dataset. After training, the network is tested using the testing dataset and it predicts whether the software project classes have the code clones or not.

1) Log sigmoid Activation function: Log sig is a non-linear transfer function used to train neural networks. It is simply called sigmoid function. It calculates the layer's output from the net input. It has the "s" shaped curve.

$$S(t) = 1/1+e^{-t}$$

IV. SIMULATION OF NETWORKS

Now we test and validate the neural network using optimized testing dataset to get properly weighted & biased neural networks. Now using these trained networks we supply test dataset with known target values and record the output as existence of clone in the particular classes.

V. RESULT AND DISCUSSION

Results shows better accuracy and also depict the performance parameters that is precision, recall, FAR and FRR.

Table 1: Test case in False Acceptance Rate with clone detection

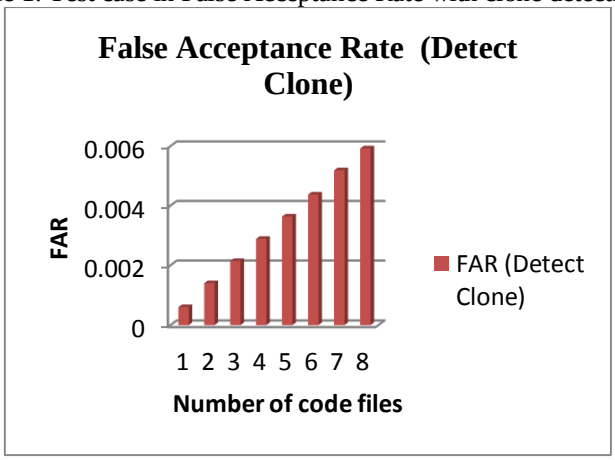


Table 2: Test case in False Rejection Rate with clone detection

No. of Code Files	FRR (Detect Clone)
1	0.0006
2	0.00132
3	0.00205
4	0.0027
5	0.0035
6	0.0042
7	0.0050
8	0.00582
9	0.00656
10	0.007186

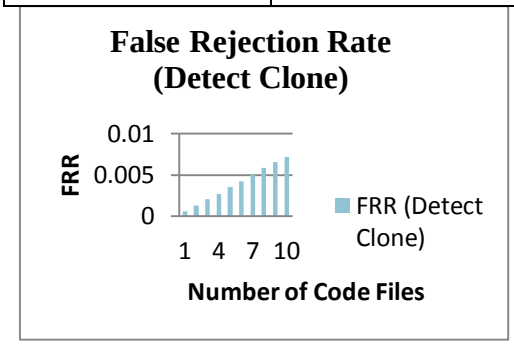


Fig 2. Test Case between FRR (Detect Clone)

The above figure represents that the comparison between the false rejection rate with clone and not clone case. We identified the wrong data rejected the case, the clone detection case rejected data are less. We improve the performance of the wrong data is rejected in detection case with ACO and BPNN

Table 3: Test case in Accuracy with clone detection

No. of Code files	Accuracy (Detect Clone)
1	8.9
2	18.91
3	28.87
4	39.81
5	48.77
6	57.73
7	67.68
8	78
9	89
10	97

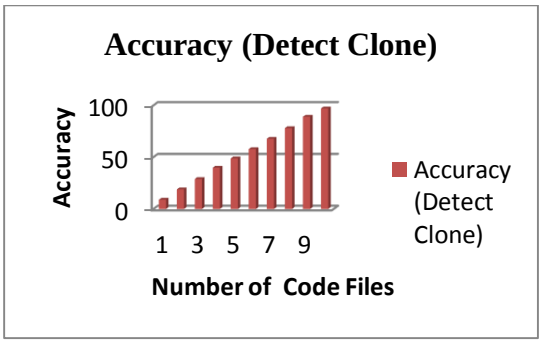


Fig 3. Test Case between Accuracy (Detect Clone)

Above figure defines the accuracy based on false acceptance rate and rejection rate. Far and Frr is minimum ration then improve the performance of the detection software tool. In clone case we achieve the accuracy with BPNN value is 98.54 and not clone case, we achieved the Frr values are 92.02.Table 4: Test case in Precision with clone detection

No. of Code Files	Precision (Detect Clone)
1	0.04569
2	0.09645
3	0.1472
4	0.198
5	0.248
6	0.299
7	0.350
8	0.401
9	0.4569
10	0.4975

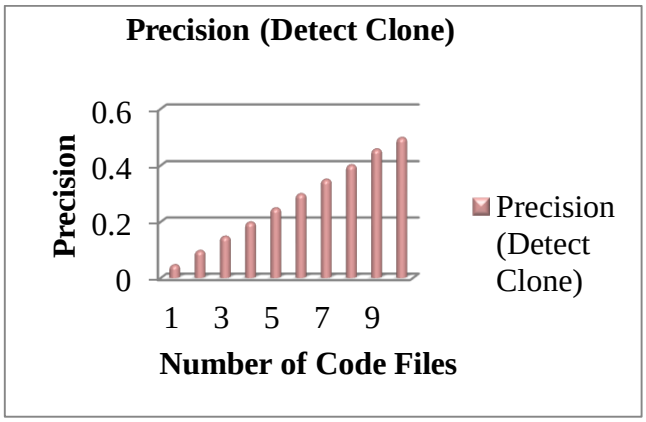


Fig 4. Test Case between Precision (Clone Detection)

Figure define precision is also called positive predictive value is the fraction of retrieved instances that are relevant. In clone case we achieve the precision with BPNN value is 0.5025 and not clone case, we achieved the precision values are 0.4953.We improve the performance of the precision value clone detection. We improve the performance of the precision value in not clone detection with ACO and BPNN.

Table 5: Test case in Recall with clone detection

No. of Code Files	Recall (Detect Clone)
1	0.000735
2	0.001324
3	0.002133
4	0.002868
5	0.0036
6	0.0044
7	0.005
8	0.0058
9	0.00661
10	0.0072

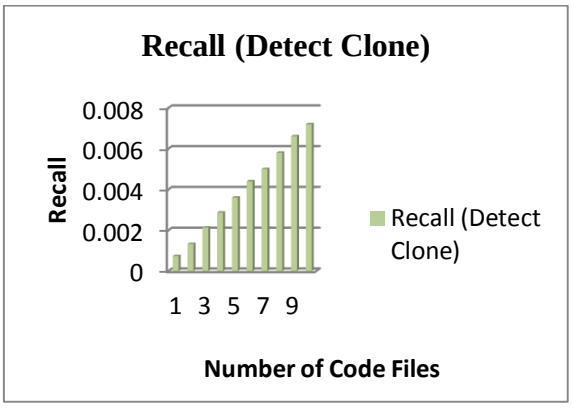


Fig 5. Test Cases between Recall (Clone Detection)

Figure recall is also known as sensitivity is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. In clone case we achieve the Recall with BPNN value is 0.5025 and not clone case, we achieved the recall values are 0.4953.We improve the recall value in clone detection case with the help of BPNN.

VI. CONCLUSION

We observed that a multilayer back propagation neural network with activation function is much suitable for the prediction of code clone using the metric based optimized dataset and we also observed that ant colony optimization technique is best to generate optimized dataset for reduce the complexity of given dataset. Moreover, in this study we calculate some performance parameters that is FAR (false acceptance rate), FRR (false rejection rate), Precision, Recall and Accuracy. This study can be further extended for the identification of types of clones. The overall performance of the dataset generated by the tool is good and hence the analysis of code clone detection using software metrics and neural network is much more effective technique as compared to regression analysis and other clone detection techniques. Back propagation neural network has the immense capability of data prediction. Prediction and identification of clones using BPPN can decrease the maintenance cost and hence increase the software quality which leads to faster software development.

VII. REFERENCES

[1]. Roy CK, Cordy JR, Koschke R. "Comparison and evaluation of clone detection techniques and tools: A qualitative approach. Science of Computer Programming "2009; 74:470–495.
[2]. Roy CK, Cordy JR. A survey on software clone detection research. TR 2007-541, Queen's School of Computing, 2007; 115
[3]. D.Gayathri Devi, Dr. M. Punithavalli "Comparison and Evaluation on Metrics based Approach for Detecting Code Clone" in IJCSE, ISSN: 0976-5166 Vol. 2 No. 5 Oct-Nov 2011 page no- 750.
[4]. Pavitdeep Singh, Prof. Satwinder Singh, Prof. JatinderKaur "Tool for Generating Code Metrics for C# Source Code using Abstract Syntax Tree Technique" ACM Sigsoft, software engineering notes-vol-3 sep-2013 pages1-6
[5]. Sandeep Sharawat : Software maintainability prediction using Neural Network, Vol. 2, Issue 2,Mar-Apr 2012, pp.750-755
[6]. Kodhai. E, Perumal. A, and Kanmani. S,"Clone Detection using Textual and Metric Analysis to figure out all Types of Clones", in IJCCIS, Vol2. No1. ISSN: 0976–1349 July-Dec 2010.
[7]. RubalaSivakumar, Kodhai. E,"Code Clones Detection in Websites using Hybrid Approach", in IJCA (0975 – 888) Volume 48–No.13, June 2012.
[8]. Dr. C.R.K Reddy, Dr. A.goverdhan and G.Anilkumar, "An efficient method-level code clone detection scheme through textual analysis using metrics" IJCET, ISSN: 6375, Vol-3, Issue-1, Jan-June 2012, pp 273-288.
[9]. Yogita Sharma, Rajesh Bhatia, "Hybrid technique for object oriented software clone detection'- a thesis submitted in june-2011, Thapar University, Patiala.
[10]. Mie MieThetThwin, Tong-SengQuah" Application of Neural Networks for Software quality Prediction Using Object-Oriented Metrics" in ICSM, ISSN: 1063-6773 Sep-2003.
[11]. Filip Van Rysselberghe, Serge Demeyer. Evaluating Clone Detection Techniques. In Proceedings of the International Workshop on Evolution of Large Scale Industrial Applications (ELISA'03), 12pp., Amsterdam, The Netherlands, Sept 2003.
[12]. Dr. C.R.K Reddy, Dr. A.goverdhan and G.Anilkumar, "An efficient method-level code clone detection scheme through textual analysis using metrics" IJCET, ISSN: 6375, Vol-3, Issue-1, Jan-June 2012, pp 273-288.
[13]. R. Koschke, R. Falke, P. Frenzel, Clone detection using abstract syntax suffix trees, in: Proceedings of the 13th Working conference on Reverse Engineering, WCRE 2006, 2006, pp. 253–262.
[14]. B. Baker, A prgram for identifying duplicated code, in: Proceedings of Computing Science and Statistics: 24th Symposium on the Interface, vol. 24, 1992, pp. 49–57.
[15]. B. Baker, On finding duplication and near-duplication in large software systems, in: Proceedings of the 2nd Working Conference on Reverse Engineering, WCRE 1995, 1995, pp. 86–95.
[16]. Sukhpreet kaur, Prof. Satwinder singh , " Code clone detection and analysis using software design object oriented metrics" IJCET, ISSN 2319-7080, Vol-4, Issue-3, Nov 2015.