ISSN: 2395-3519

International Journal of Advanced Trends in Computer Applications
www.ijatca.com

CODE CLONE DETECTION USING WEIGHTED FEED FORWARD BACK PROPOGATION NEURAL NETWORK

[1]Er. Rajnish Bala, [2]Er. Navpreet Rupal

[1]MTech (CSE)
Shaheed Udham Singh College of Engg. & Technology, Tangori
[2]Asst. Prof
Shaheed Udham Singh College of Engg. & Technology, Tangori
[1]rajnijindal27@gmail.com,[2] er.nrupal@gmail.com

Abstract: *This research studies the various clone detection methods to propose the novel clone detection technique. Clone Detection first parses the source code and then performs data mining with plan on the parsed code. All similar code segments are recognized and then irregularity recognition is performed. The whole process is executed in an issue of minutes. Outcomes are stored in a database for future use. Neural Network, SVM, Data mining are used to evaluate the performance of the system and the parameters FAR, FRR and accuracy are used.*

Keywords: *Code clone detection, SVM, Neural Network, Data mining, FAR, FRR, Accuracy.*

1. INTRODUCTION

Clones typically occur when a phrase is copied and optionally shortened, producing exact or near miss clones. Though, code sections that are parallel but not matching arise repeatedly in performance, finding such non- equal clones can be as vital as finding matching code segments. Like, as automated code compaction might need ruling the same clones, study of the development of a code base over time need judging clones that differ in their resemblance. One of the vital issues with ruling non-identical clones is executing when two pieces of code are close enough to be measured "alike". As, this is expect to depend on the situation in which the clone recognition instrument is used, it is believe that such tools should supply a quantitative measure of clone resemblance, parting the final choice of categorization to the user of the device. Detecting code clones in a code base is a very tricky. As per open source and commercial code, 66% of cloned code is customized [3-6]. Clone Detection first parses the source code and then performs data mining with plan on the parsed code. All similar code segments are recognized and then irregularity recognition is performed. The whole process is executed in an issue of minutes. Outcomes are stored in a database for future use. A web based interface supports efficient performance and study of all the detected code clones with possible defects [8] [10].

1.2 Why code clone detection is essential?

Code Clone Detection is not like other clone detection tools. Here is a summing up of the key differences [15-17]:

- It's quicker
- It's more absolute
- It detects bug
- It's easier to draw on
- It includes project parameters
- It's an open platform

1.3 Clone Types

There are two main kinds of link between code fragments. Fragments can be alike as on the likeness of their program text or they can be alike based on their functionality. The first type of clone is often the effect of repetition a code section and pasting into another location. In the following, the provision of types of clones based on both the textual is given (Types 1 to 3) [3] and functional (Type 4) [5, 6] similarities:

A. T ype-1

Identical code fragments apart from variations in whitespace, layout and comments.

B. Type-2
Syntactically identical fragments apart from variations in identifiers, literals, types, whitespace, layout and comments.

C. Type-3
Unoriginal fragments with additional modifications like changed, added or removed statements with variations in identifiers, literals, types, whitespace, layout and comments.

D.Type-4
Two or more code fragments that execute the similar calculation but are implemented by diverse syntactic variants would be considered.

1.4 Detection Techniques [9]

A. String Based
String based techniques utilize necessary string alteration and contrast algorithms that makes them free of programming languages. Techniques in this group vary in essential string comparison algorithm. Comparing intended signatures per line is one option to recognize for identical substrings. Line matching that comes in two variants is an option that is selected as an agent for this group as it utilizes common string manipulations.

B. Token Based
Token based techniques apply a more complicated alteration algorithm by constructing a token stream from the source code, therefore need a lexer. The existence of such tokens makes it feasible to use enhanced comparison algorithms. Then to parameterized matching with suffix trees, that behaves as representative.

C. Parse tree Based
Parse tree based techniques exercise a heavyweight transformation algorithm, i.e. the building of a parse tree. Because of the richness of this structure, it is possible to try various comparison algorithms as well.

2. Data Mining

Data mining derives its name from the similarity among searching for precious information in a large database and mining rocks for an element of valuable ore. Together they involve either sifting during a huge quantity of substance or inventively probing the substance to precisely pinpoint where the values exist in. It is, though, a misnomer, as mining for gold in rocks is usually called “gold mining” and not “rock mining”, therefore by analogy, data mining be supposed to called “knowledge mining” instead. But, data mining become the conventional customary term and very quickly a tendency that even overshadowed more general terms such as knowledge discovery in databases (KDD) that explain extra whole process.
Data mining is being put into use and calculated for databases, with relational databases, object-relational databases and object oriented databases, data warehouses, transactional databases, unstructured and semi structured repositories like World Wide Web, advanced databases such as spatial databases, multimedia databases, time-series databases and textual databases, and even flat files [25] [27].

3. Neural Network

Machine learning algorithms facilitate a lot in decision making and neural network has performed well in categorization purpose in medical field. Most popular techniques among them are neural network. Neural networks are those networks that are the collection of simple elements which function parallel. A neural network can be trained to perform a particular function by adjusting the values of the weights between elements. Network function is determined by the connections between elements. There are several activation functions that are used to produce relevant output [26].

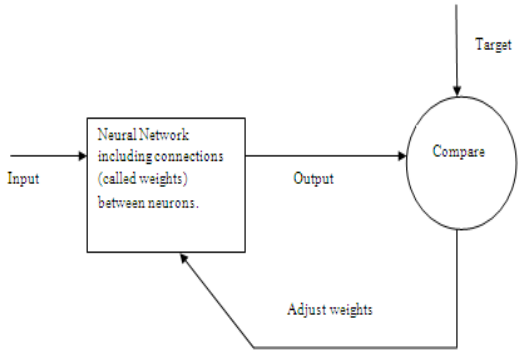

Figure 1: Neural Network Block Diagram

Training can be either supervised or unsupervised. In supervised training system learns by trying to predict outcomes for known examples. System compares its predictions with the known results and learns from is mistakes. In unsupervised training system no output or result is shown as part of training process. With the delta rule, as with other types of back propagation, learning is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of output, and the backward error propagation of weight adjustments. Simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it may be. It then sees how far its

respond was from the actual one and makes unsuitable adjustment to its relationship weights. Within each hidden layer node is a sigmoid activation function which polarizes network action and helps it to be regular in nature. Back propagation performs a slope drop within the solution's vector space towards a 'global minimum' along the steepest vector of the mistake surface. The global minimum is that make up solution with the lowest probable error. Back propagation is a technique of training artificial neural network. It requires an unwanted output for each value in order for computation of loss function gradient. Following algorithm will show how BPNN works in classification in medical imaging. Chiefly the error back propagation process consists of two passes from side to side the dissimilar layers of the network a forward pass and a diffident pass.

4. SVM (Support Vector Machine)

SVMs introduced in COLT-92 by Boser, Guyon & Vapnik and became rather accepted since. Support Vector Machine (SVM) is mostly a classier method that performs categorization tasks by constructing hyper planes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle several continuous and categorical variables. For categorical variables a dummy variable is shaped with case values as either 0 or 1. Therefore, a categorical dependent variable consisting of three levels, say (A, B, C), is represented by a set of three dummy variables.

A :{ 1 0 0}; B {0 1 0}; C {0 0 1}

To build an optimal hyper plane, SVM employs an iterative training algorithm, which is used to minimize an error function.

5. IMPLEMENTATION PROCESS

For the better accuracy of this work, the steps followed by uploading the source code for initializing the neural network and its hidden layer to find the clone. After that, the comparison of various parameters with respect to Support Vector Machine and Neural Network is done. The para metrics are FAR, FRR and accuracy.

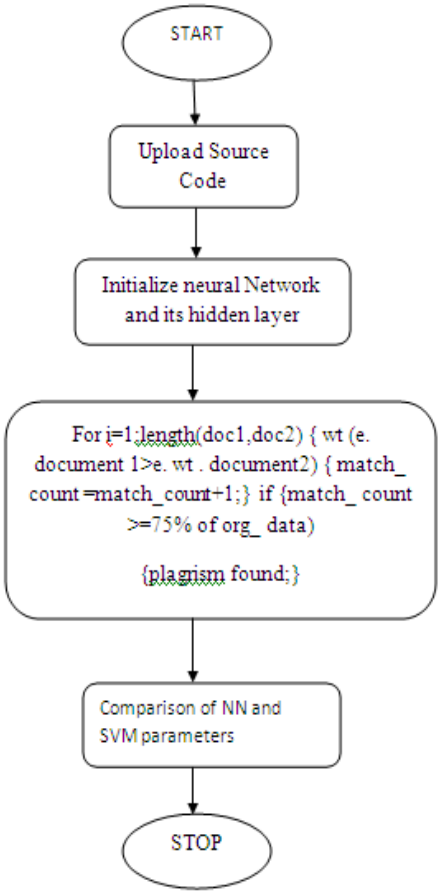

Figure 2: Methodology of Code clone detection

6. RESULTS AND DISCUSSION

The reliability of the proposed code clone detection is described with the help of experimental results. The training data contains FRR, FAR and Accuracy parametric using SVM and Neural networks.

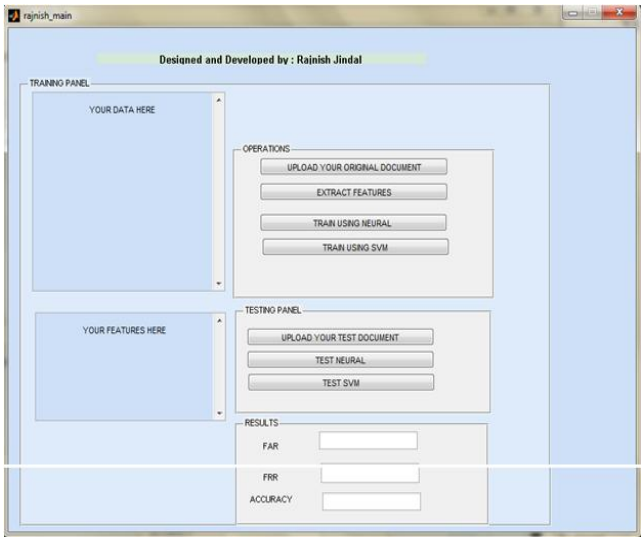

Figure 3: GUI uploading phase

In this ,after uploading of original document, extraction of feature using Generator function. After this process, go for training part firstly through neural network and

secondly by support vector machine (SVM). In the testing panel, uploadation of test document . After this process, go for testing part firstly through neural network and secondly by support vector machine (SVM).

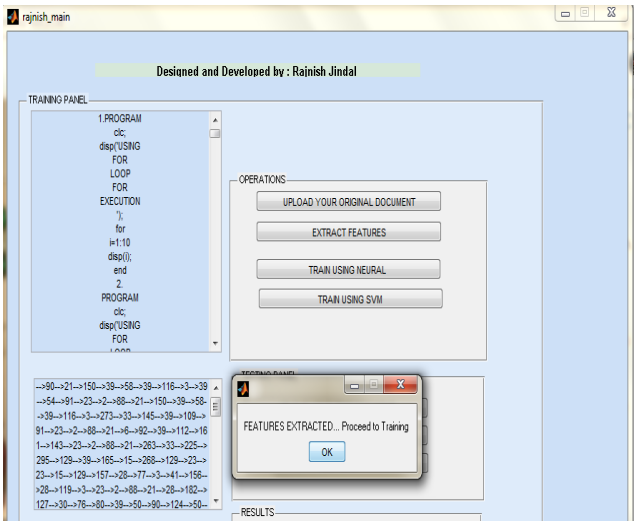

Figure 4: Feature Extraction (using generator function)

In this figure, uploading of original data in the form of source code. extraction of feature using Generator function. After the training, a box contains message would appear then go for training part firstly through neural network and secondly by support vector machine (SVM). In the testing panel, uploadation of test document.

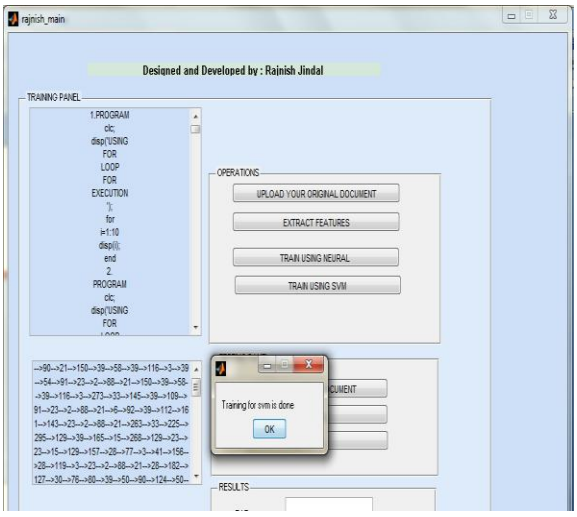

Figure 5: Training phase (NN and SVM)

Uploading of original document then the extraction of features of original document would be done. Then the training using neural network and with SVM will be done.

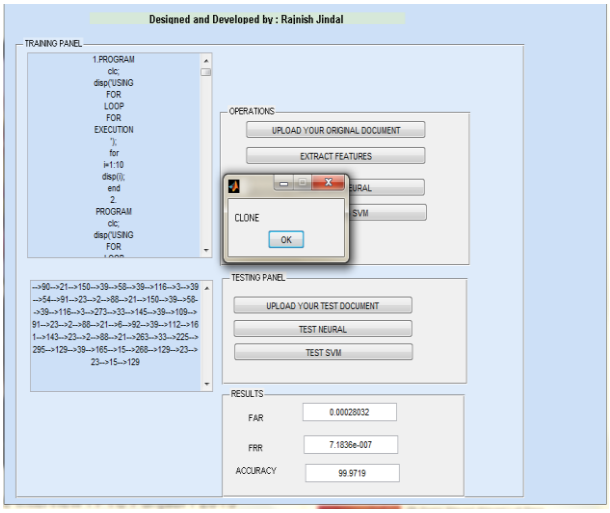

Figure 6: Testing phase

In this figure, testing part is considered. First of all, upload of test document in form of source code, then testing with the neural and then with the SVM. And while testing, accuracy will be 99.1% in case of Neural and in case of SVM, the accuracy will be 84.7%.

Iterations	FAR	FRR	Accuracy
10	0.0014	0.03	96.86
20	0.0015	0.08	91.85
30	0.0014	0.11	99.88
40	0.0017	0.05	94.83
50	0.0018	0.02	97.82

Table 1: Neural network values w.r.t to FRR, FAR and Accuracy

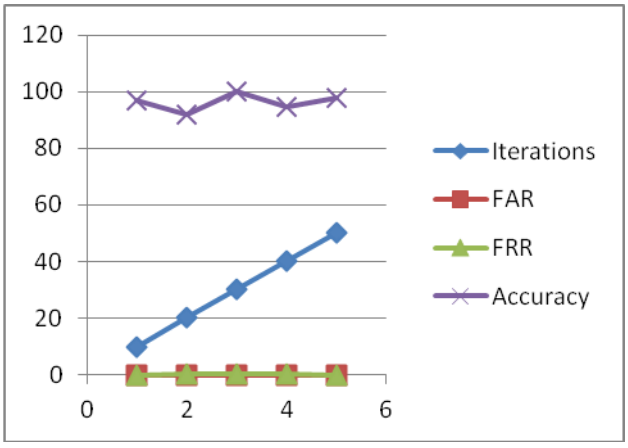

Figure 7 : Parametrics of Neural Network

The above figure shows the false acceptance rate and false rejection rate using neural network with respect to the number of iterations and shows that these performance parameters is having less measure which is having high accuracy on the basis of false acceptance rate and false rejection rate and shows that neural

network classifier performance for the code colon detection.

Iterations	FAR	FRR	Accuracy
10	0.0018	0.09	90.82
20	0.002	0.11	87
30	0.0019	0.17	82.81
40	0.0025	0.19	80.75
50	0.0022	0.09	90.78

Table 2: SVM values w.r.t to FRR, FAR and Accuracy

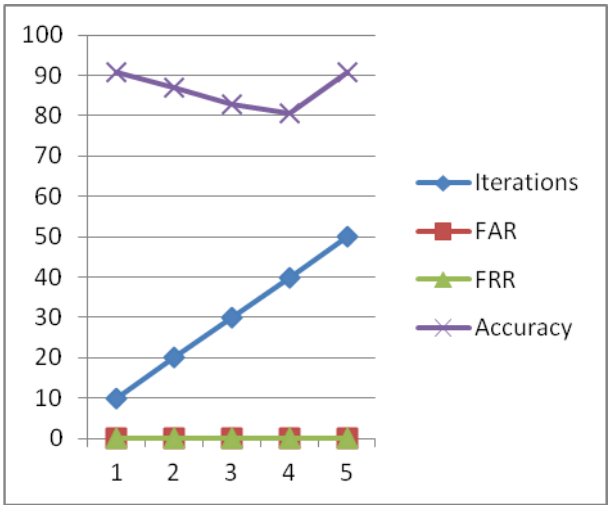

Figure 8: Para metrics of SVM

The figure 6.3.2 shows the false acceptance rate and false rejection rate using Support Vector Machine (SVM) with respect to the number of iterations and shows that these performance parameters having less accuracy on the basis of false acceptance rate and false rejection rate and shows that SVM classifier performance for the code colon detection , that is the performance of Neural Network classifier is better than SVM.

7. CONCLUSION

The work deals with improving the efficiency by using Support Vector Machine and Neural Network following various parameters that are FAR, FRR and Accuracy. Accuracy is the main constraint for researchers on computer system as accuracy would be more if FRR and FAR would be less. The results have been compared of SVM and NN.SVM doesn't predict classification. SVM shows less accuracy as compare with Neural Network that has more iterations and a training model that helps neural networks for better results.

8. FUTURE SCOPE

It is shown that the proposed work radically improves accuracy of neural network with the help of training model and enables the system to meet system requirements. In future, feature generation system can be used that can be updated by Genetic algorithm, BFO technique that can radically lower the values of FAR and FRR.

REFERENCES

[1].Anna Bartkowiak, "Neural Networks and Pattern Recognition," Institute of Computer Science, University of WrocAlaw.

[2].Amandeep Kaur, Mandeep Singh Sandhu**, "**Software code clone detection model using hybrid approach", in IJCT, Volume 3 No.2, OCT, 2012.

[3]. Blawinder Kumar, "Analysis of Code Clone Detection using Object Oriented System and Neural Network", International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 9, September- 2014.

[4]. Chanchal K. Roy, James R. Cordy, "Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach," Preprint submitted to Science of Computer Programming February 24, 2009.

[5]. Chatterji, "Effects of cloned code on software maintainability: A replicated developer study", Reverse Engineering (WCRE), 2013 20th Working Conference, IEEE, 2013.

[6].Dr.Gayathri Devi, Dr. M. Punithavalli, "Comparison and Evaluation on Metric based approach for detecting code clone", Indian Journal of Computer Science and Engineering, Vol. 2 No. 5 Oct-Nov 2011.

[7].Dmitriy Fradkin and Ilya Muchnik, "Support Vector Machines for Classification," 2000 Mathematics Subject Classi_cation. 62H30.

[8]. Er. Rajnish Bala, Er. Navpreet Rupal, "A survey on clone detection and clone analysis," International Journal of Advanced Trends in Computer Applications (IJATCA) Volume 1, Number 5, May - 2015, pp. 17-20 ISSN: 2395-3519.

[9].Filip Van Rysselberghe, Serge Demeyer, "Evaluating Clone Detection Techniques," http://www.refactoring.com/ for an overview of IDE's supporting refactoring.

[10].Girija Gupta, Indu Singh, "A Novel Approach Towards Code Clone Detection and Redesigning", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013 ISSN: 2277 128X.

[11].Hassan Raza Bhatti, "Automatic Measurement of Source Code Complexity", Master's Thesis submitted at Luleå University of Technology, 2011.

[12]. Hiroaki Murakami, Keisuke Hotta, "Folding Repeated Instructions for Improving Token-based Code Clone Detection," 2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation.

[13]. Ke Wang, Philip S. Yu, "Bottom-Up Generalization: A Data Mining Solution to Privacy Protection," Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04) 0-7695-2142-8/04 $ 20.00 IEEE.

[14].Kodhai. E, Kanmani. S, Kamatchi. A, Radhika. R, "Detection ofType-1 and Type-2 Clone Using Textual Analysis and Metrics", in ITC, 2010 IEEE..

[15].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.

[16].Kodhai, SelvaduraiKanmani, "Method-level code clone detection through LWH (Light Weight Hybrid) approach",Journal of Software Engineering Research and Development, October 2014, 2:12.

[17].Leonardo Moura, Marcelo Sant'Anna, "Clone Detection Using Abstract Syntax Trees," Copyright 1998 IEEE. Published in the Proceedings of ICSM'98, November 16-19, 1998.

[18].Liu Yucheng, "Incremental Learning Method of Least Squares Support Vector Machine," 2010 International Conference on Intelligent Computation Technology and Automation.

[19].Mohammed Abdul Bari, Dr.ShahanawajAhamad, "Code Cloning: The Analysis, Detection and Removal", in InternationalJournal of Computer Applications (0975 – 8887), Volume 20–No.7, April 2011.

[20].Mustafa Kapdan, Mehmet Aktas, MelikeYigit, "On the Structural Code Clone Detection Problem: A Survey and Software Metric Based Approach"Computational Science and Its Applications – ICCSA 2014Lecture Notes in Computer Science Volume 8583, 2014, pp 492-507.

[21].Osmar R. Zaïane, "Introduction to Data Mining," Osmar R. Zaïane, 1999 CMPUT690 Principles of Knowledge Discovery in Databases.

[22].Prajila Prem, "A Review on Code Clone Analysis and Code Clone Detection," ISSN: 2277-3754 ISO 9001:2008 Certified International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 12, June 2013.

[23]. RubalaSivakumar, Kodhai. E,"Code Clones Detection in Websites using Hybrid Approach", in IJCA (0975 – 888) Volume 48–No.13, June 2012

[24].Savvas Karatsiolis, "Region based Support Vector Machine Algorithm for Medical Diagnosis on Pima Indian Diabetes DataSet," Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012.

[25].Salwa K. Abd-El-Hafiz, "A Metrics-Based Data Mining Approach for Software Clone Detection", 2012 IEEE 36th International Conference on Computer Software and Application.

[26].Wouter gevaert, georgi tsenov, "Neural networks used for speech recognition," Journal of automatic control, university of belgrade, VOL. 20:1-7, 2010.

[27].Xindong Wu, Xingquan Zhu, "Data mining with Big Data," Digital Object Indentifier 10.1109/TKDE.2013.109 1041-4347/13/$31.00 © 2013 IEEE.

[28].Yogita Sharma, "Hybrid technique for object oriented software clone detection", M.E Thesis submitted at Thapar University, Patiala, 2011.

[29].Yoshiki Higo, Ken-ichi Sawa "Problematic Code Clones Identification using Multiple Detection Results," 2009 16th Asia-Pacific Software Engineering Conference.

[30].Yoshiki Higo, Yasushi Ueda, "Incremental Code Clone Detection: A PDG-based Approach," 2011 18th Working Conference on Reverse Engineering