

Software Patterns

Applying Reverse Engineering Techniques to
Analyze Design Patterns in Source Code

Introduction

- Design patterns play an important role during analysis and design phases of the development process.
- Forward engineering
- Reverse engineering

- Software design patterns are applied very often in forward engineering, where the solution of patterns is based on its candidate classes.
- Reverse engineering technique can be used to analyse the availability of patterns in the object-oriented software.

- We use object oriented metrics for defining individual participants of patterns that exist in the code of the target project.
- The use of software metrics helps to suppress the case of false negatives by adapting various applications of design patterns and by distinguishing design patterns where the structure of classes are identical.

Introducing Learning

- We can also map the design pattern analysis process into the process of learning
- Supervised machine learning techniques, like Artificial Neural Networks (ANN) can be used for the process of pattern detection.

Related Work

- Gueheneus, H and F. Zaidi have proposed a manual pattern analysis technique in their paper “Fingerprinting design patterns” in 2004.
- With this technique, they produced a pattern repository which other techniques often use for the process of validation.

- Authors then started analysing various open source software and have applied ML algorithms to detect patterns.
- A software metrics based detection was proposed “Applying software metrics for the mining of design pattern” in 2016 where the JUnit software is considered as a case study.
- 2 learning based methods namely the Layered Recurrent Neural Network and Random Forest have been used to classify patterns

- Uchiyama et al. have also presented a learning-based method for the analysis of design patterns in their paper “Design pattern detection using software metrics and machine learning”.
- Authors have determined Adapter and Strategy design patterns by using two ML models back propagation and decision tree.
- They have also considered StarWriter as a case study for analyzing Adapter and Strategy design patterns.

- Alhusain et al. have proposed the detection of design patterns that is based on ML in their paper “Towards Machine Learning based design pattern recognition”.
- Authors have considered ANN for the design detection process.

- Chihada et al. have presented a reverse engineering method based on ML technique.
- They have considered SVM for the process of pattern classification.
- They have also used 45 software metrics on JHotDraw, JRefactro and JUnit for the preparation of dataset.

Preliminaries

Software Design Patterns:

- Application considers Adapter, Bridge, Abstract Factory, Proxy and Template method for analysis.
- Adapter, Bridge and Proxy are Structural patterns.
- Abstract Factory is a creational design pattern.
- Template Method is a behavioral design patterns.

Preparation of Training Dataset

1) Software Pattern Definition

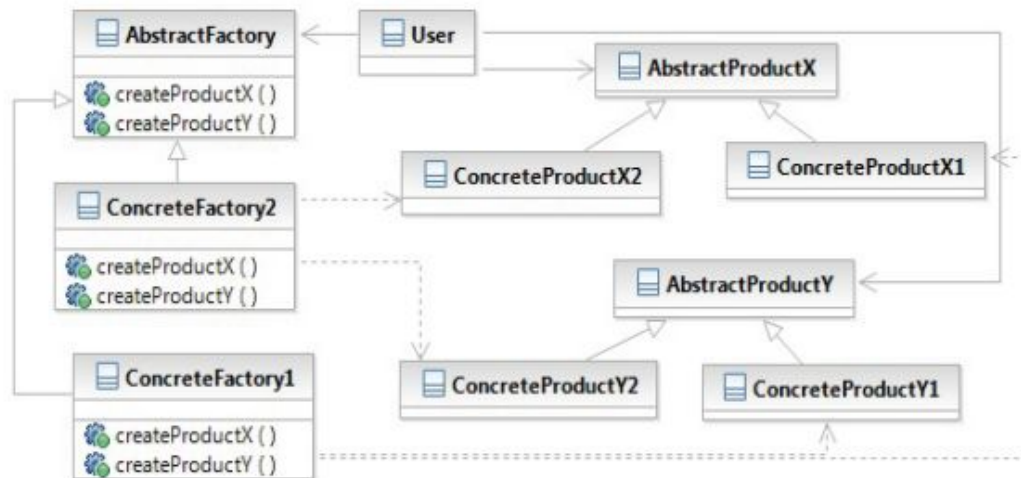
The solution of design patterns often provides high-level structure for the developed applications. The aforementioned high-level documents are abstract in nature, which lead to ambiguities in detecting design patterns. The definition of design patterns is often determined by using pattern structures as well as behavior identified by analyst.

2) Selection of Pattern's Participants

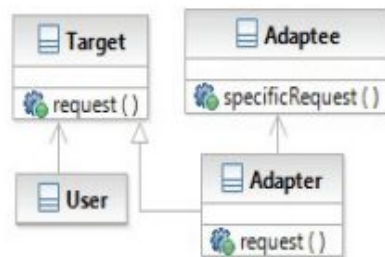
Analyst often provide input as a source code to the system, JBuilder tool for recognizing them into metrics measurements which offer measurements for each participant of a design pattern. The selected set of patterns are Adapter, Bridge, Proxy and Template Method which contain various participants. One instance is considered as multiple candidate classes (role), which need to be minimized. This process becomes helpful for increasing accuracy during pattern identification process.

3)Preparation of Object-Oriented dataset

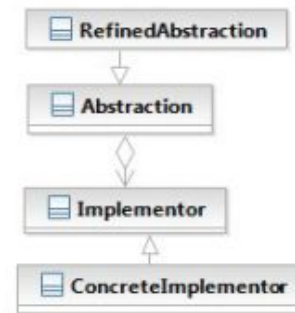
- Development of dataset is performed in parallel phases.
- JRefactory is provided as input in both phases.
- In the first phase, pattern instances are extracted from open source software whereas in second phase, metric values are extracted from source code.
- To retrieve the instances of design pattern, various pattern detection system SSA, WOP, MARPLE are considered.



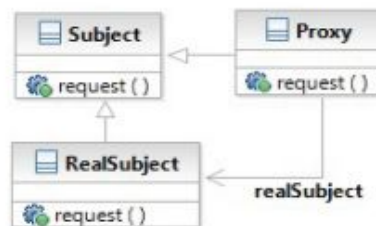
Abstract Factory



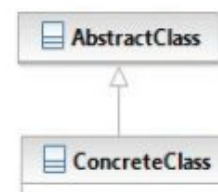
Adapter



Bridge



Proxy



Template Method

Preliminaries

List of metrics used for Pattern Analysis:

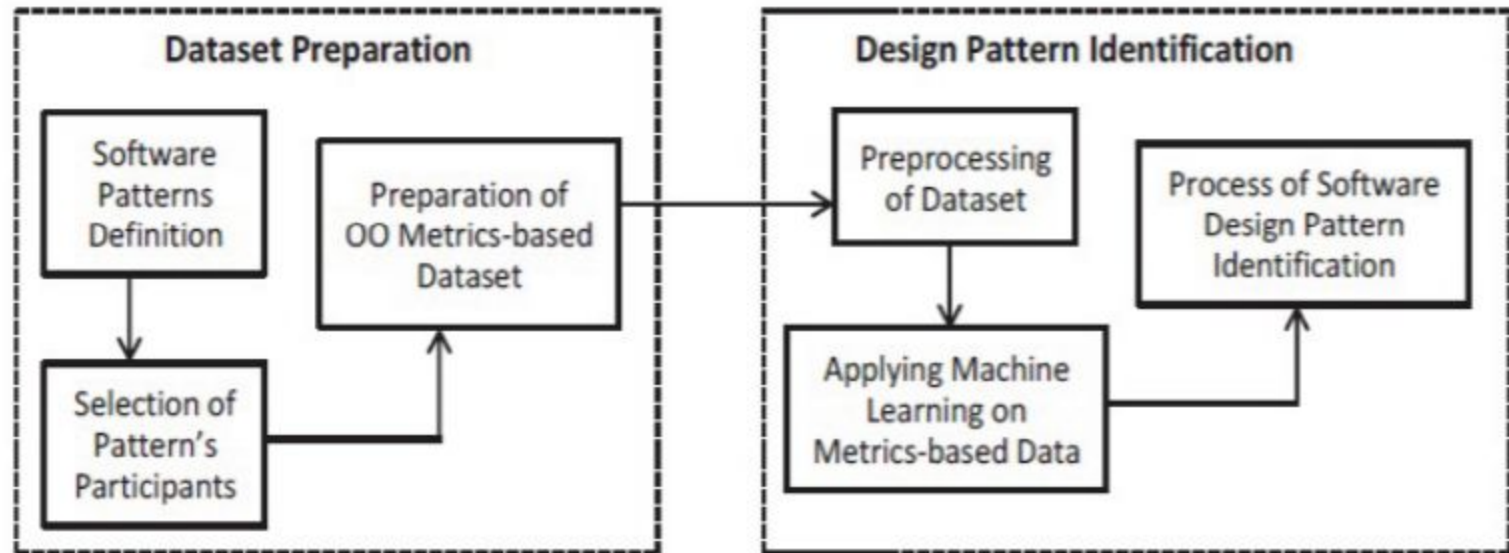
Sl.	M.	Sl.	M.	Sl.	M.	Sl.	M.	Sl.	M
1	A	15	CIW	29	IUR	43	NOA	57	PC
2	AC	16	CM	30	LCOM3	44	NOAM	58	PF
3	AHF	17	COC	31	LOC	45	NOC	59	PIS
4	AID	18	CR	32	MDC	46	NOCC	60	PS
5	AIF	19	ChC	33	MHF	47	NOCN	61	PUR
6	AIUR	20	DAC	34	MIC	48	NOED	62	RFC
7	AALD	21	DD	35	MIF	49	NOIS	63	RMD
8	AOFD	22	DOIH	36	MNOB	50	NOLV	64	TCC
9	AUF	23	EC	37	MNOL	51	NOM	65	WCM
10	CA	24	FO	38	MPC	52	NOO	66	WMPC1
11	CBO	25	HDiff	39	MSOO	53	NOOM	67	WOC
12	CC	26	HEff	40	NAM	54	NOP		
13	CE	27	HPLen	41	NCC	55	NOPA		
14	CF	28	I	42	NIC	56	NORM		

Preliminaries

Supervised Learning Method:

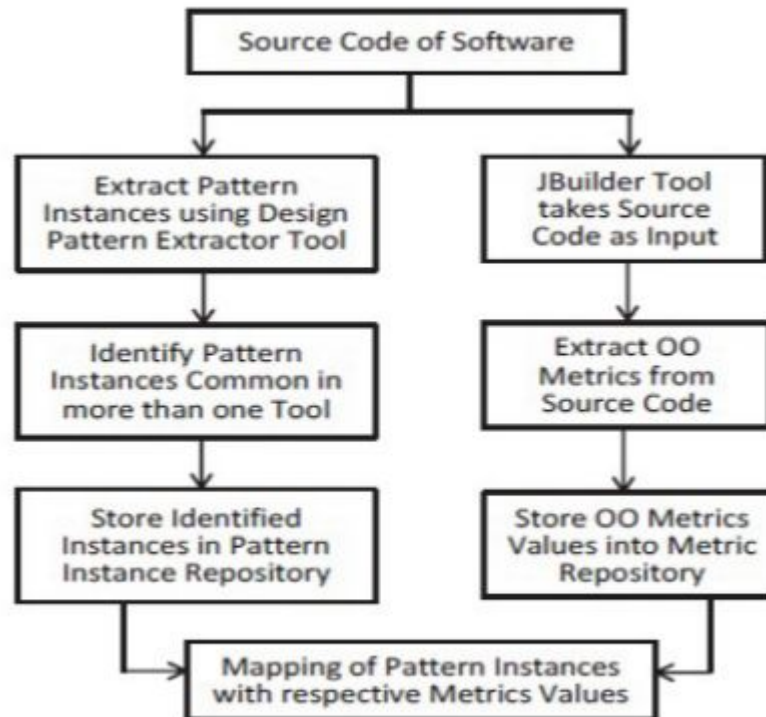
- In this study, ANN method is used for the classification process.
- A Nonlinear data modeling approach, determines complicated relationships between input and output data.
- Advantages of ANN are:
 - Includes various number of output
 - Helpful for non-linear modeling as well.
- ANN considered as random function approximation method.

Selection of Pattern's Participants



Selection of Pattern's Participants

Preparation of Object-Oriented dataset

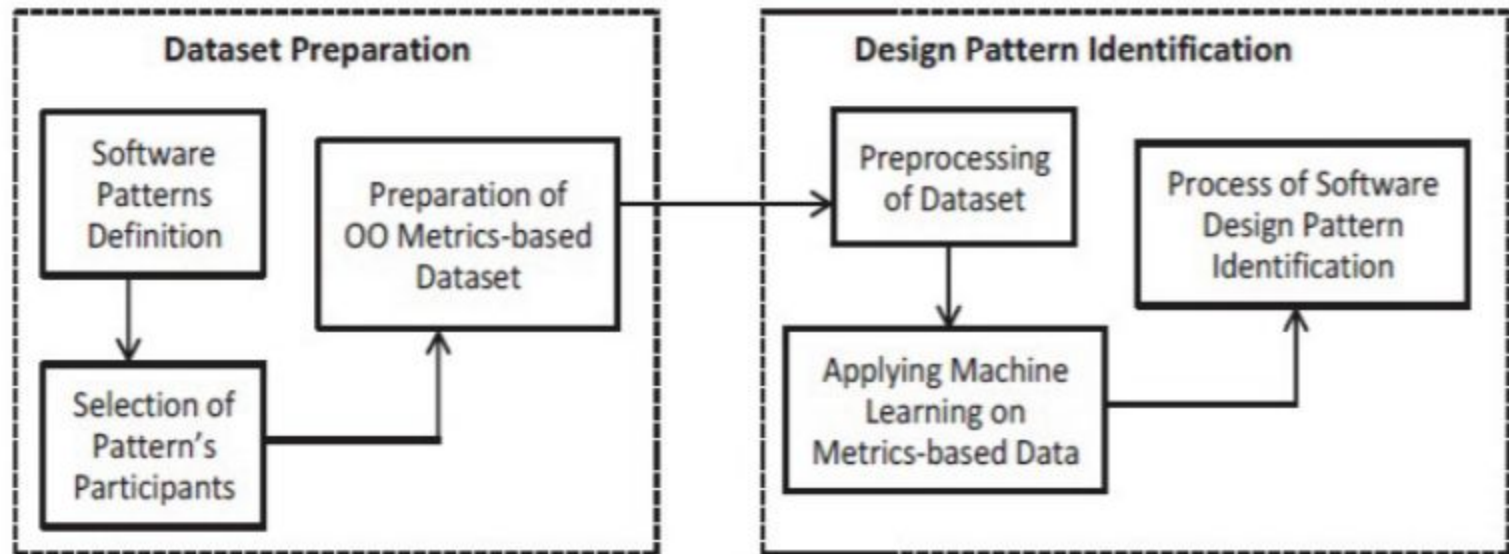


Preparation of Object-Oriented dataset

Identification of Design Patterns

The steps involved in identifying design patterns are:

- Preprocessing of dataset
- Applying Machine Learning on dataset
- Design Pattern Identification



Preprocessing of dataset

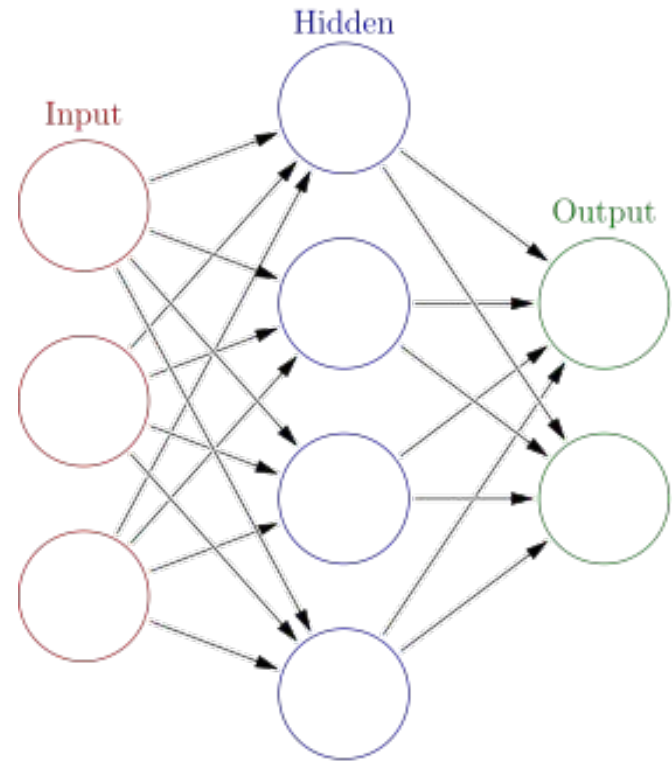
Dataset	Dimension	Notes
JRefactory	16521 x 269	268 column Input data and 1 column Target data
Quaqua	80 x 269	

Applying ML on dataset

ANN has been chosen for its ability to capture nonlinear structures with high accuracy.

The main goal is to

- Minimize false positives
- Maximize precision



Dataset

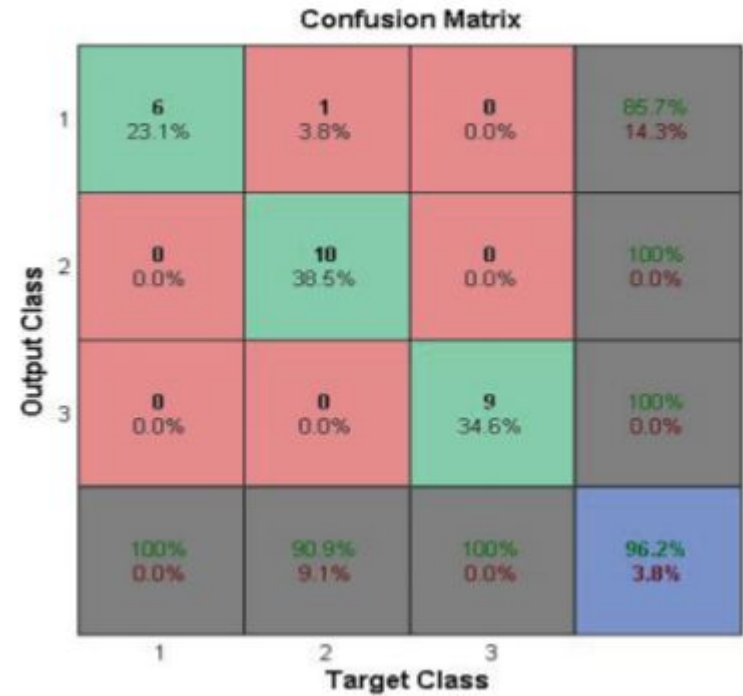
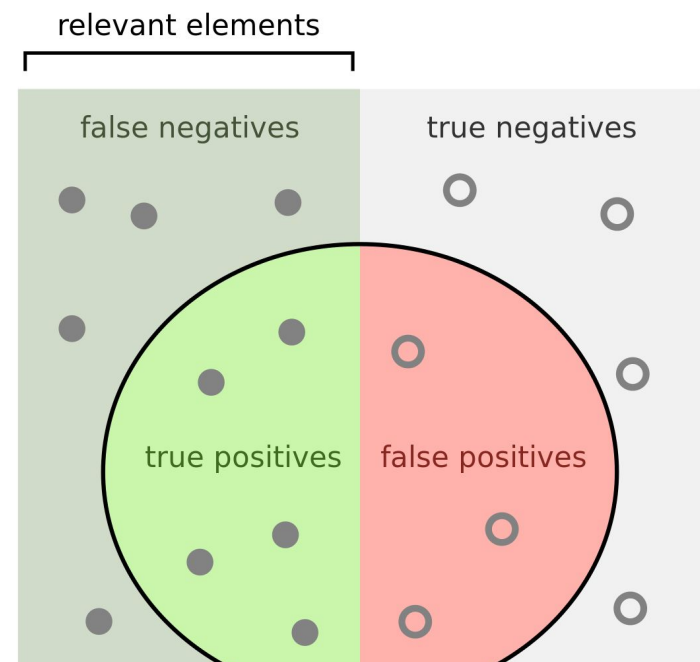
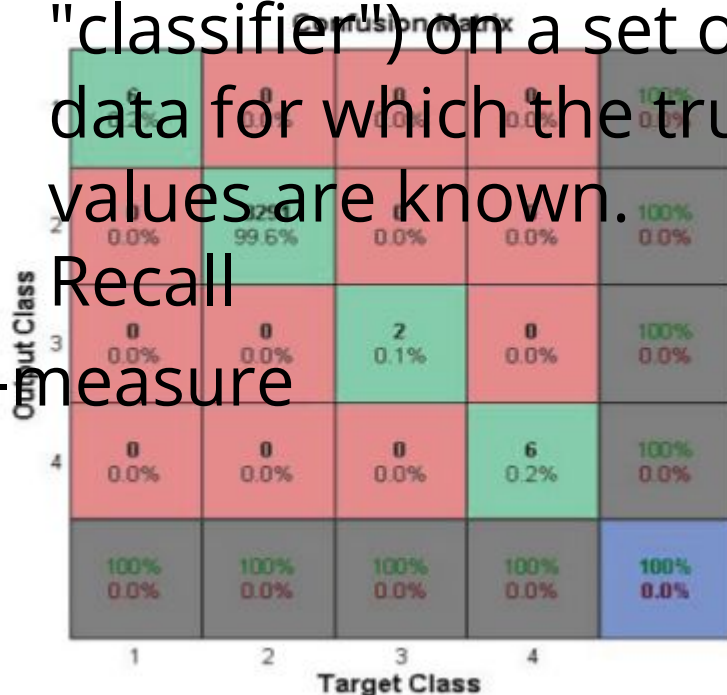
JRefractory			
Design Patterns	Pattern Instances	Training Set	Testing Set
Adapter	30	24	6
Bridge	16457	13166	3291
Proxy	10	8	2
Template Method	24	18	6

Quaqua			
Design Patterns	Pattern Instances	Training Set	Testing Set
Abstract Factory	29	23	6
Bridge	56	45	11
Template Method	45	36	9

Confusion Matrix

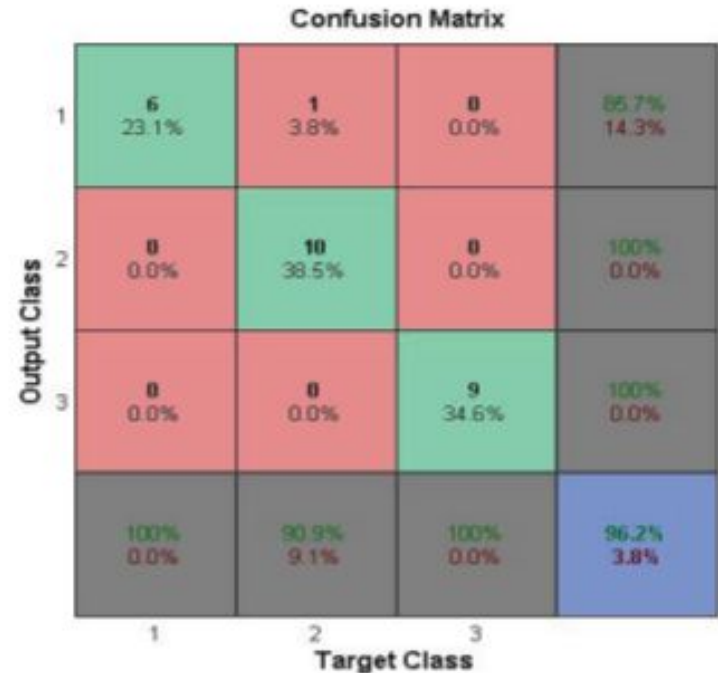
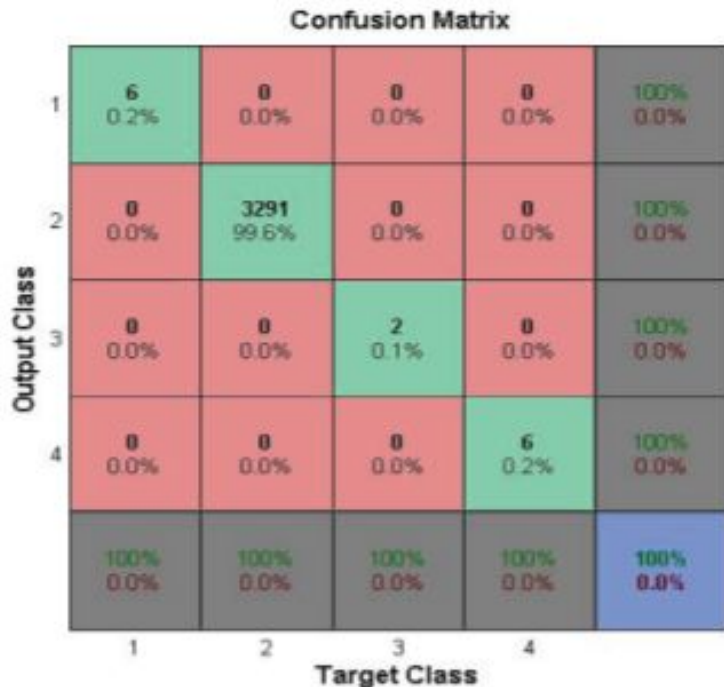
- A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Recall
F-measure



Confusion Matrix

It is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.



Accuracy

TABLE IV
RESULTS FOR JREFACTORY USING ANN

Software Patterns	Precision	Recall	F-measure	Accuracy
Adapter	100%	100%	100%	100%
Bridge	100%	100%	100%	
Proxy	100%	100%	100%	
Template Method	100%	100%	100%	

TABLE V
RESULTS FOR QUAQA USING ANN

Software Patterns	Precision	Recall	F-measure	Accuracy
Abstract Factory	100%	85.7%	92.3%	96.2%
Bridge	90.9%	100%	95.2%	
Template Method	100%	100%	100%	

Comparison.

- Pattern detection process is hard.
- The false positive rate is high.
- Not able to identify correct pattern instance, when more than one design patterns have same structure.
- Alhusain et al: Recall -92.3%
Precision -34.3%
- Tsantalis et al: Recall -100%
Precision -100%
- Uchiyama et al: Better precision and recall for small scale programs.

Comparison.

- Chihada et al: Precision -76%
Recall -76%
F-measure -76%
- SVM as machine learning method.
- Better values of F-measure during the process of pattern detection.
- Not considered Bridge and Proxy design patterns for the process of detection.

Conclusion.

- Supervised learning based on ANN..
- Design patterns: adapter, bridge, proxy, abstract factory and template method.
- JRefactory and Quaqua used for detecting these patterns.

Extension of pattern detection process to be done for other categories of design patterns.

References.

A. K. Dwivedi, S. K. Rath, S. M. Satapathy, L. S. Chakravarthy and P. K. S. Rao, "**Applying Reverse Engineering Techniques to Analyze Design Patterns in Source Code**", 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 1398-1404, doi: 10.1109/ICACCI.2018.8554519.

Pattern detection accuracy

Software Patterns	Precision	Recall	F-measure	Accuracy
Adapter	100%	100%	100%	100%
Bridge	100%	100%	100%	
Proxy	100%	100%	100%	
Template Method	100%	100%	100%	

Software Patterns	Precision	Recall	F-measure	Accuracy
Abstract Factory	100%	85.7%	92.3%	96.2%
Bridge	90.9%	100%	95.2%	
Template Method	100%	100%	100%	

