Design Patterns Detection

Gcinizwe Dlamini

Innopolis University Innopolis, Russia
g.dlamini@innopolis.university

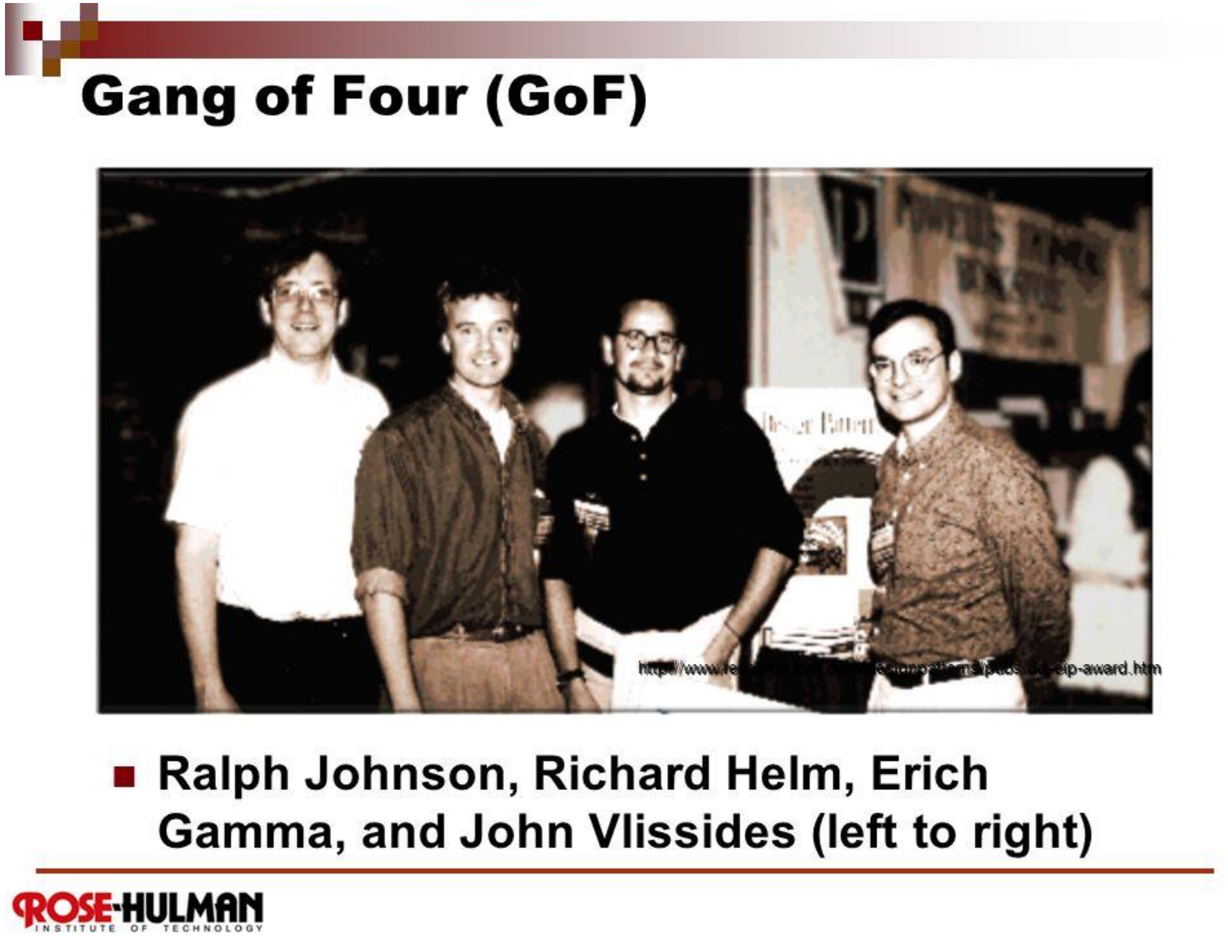
Swati Megha

Innopolis University Innopolis, Russia
s.megha@innopolis.university

Introduction

- Design Patterns Since 1994
- Gang of Four (GoF)

Introduction



Gang of Four (GoF)

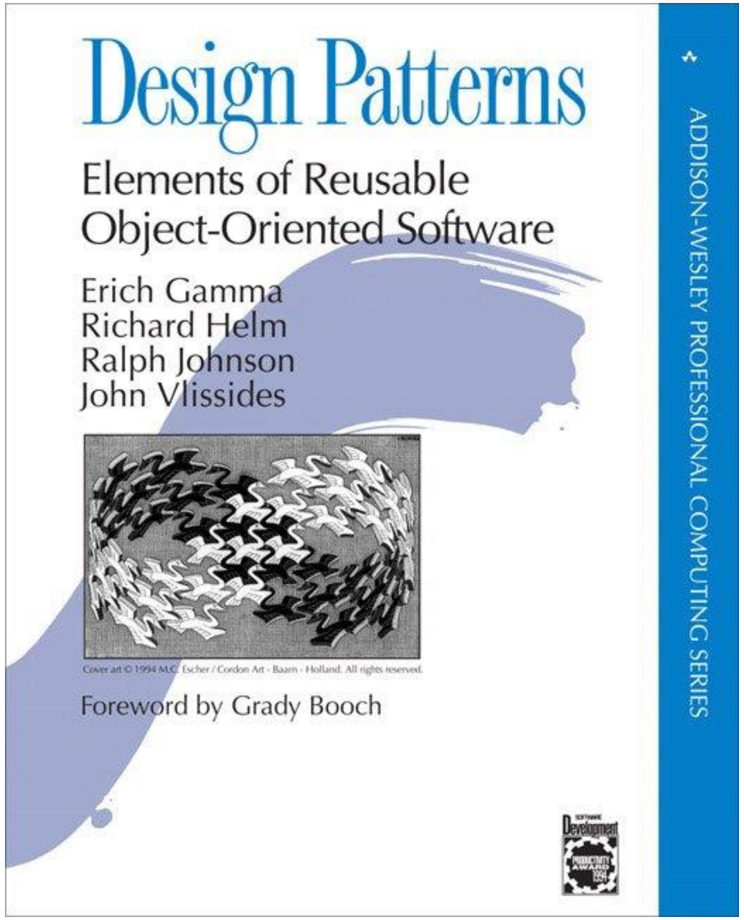
http://www.re[illegible]esignpatterns/pubs/[illegible]-eip-award.htm

- Ralph Johnson, Richard Helm, Erich Gamma, and John Vlissides (left to right)

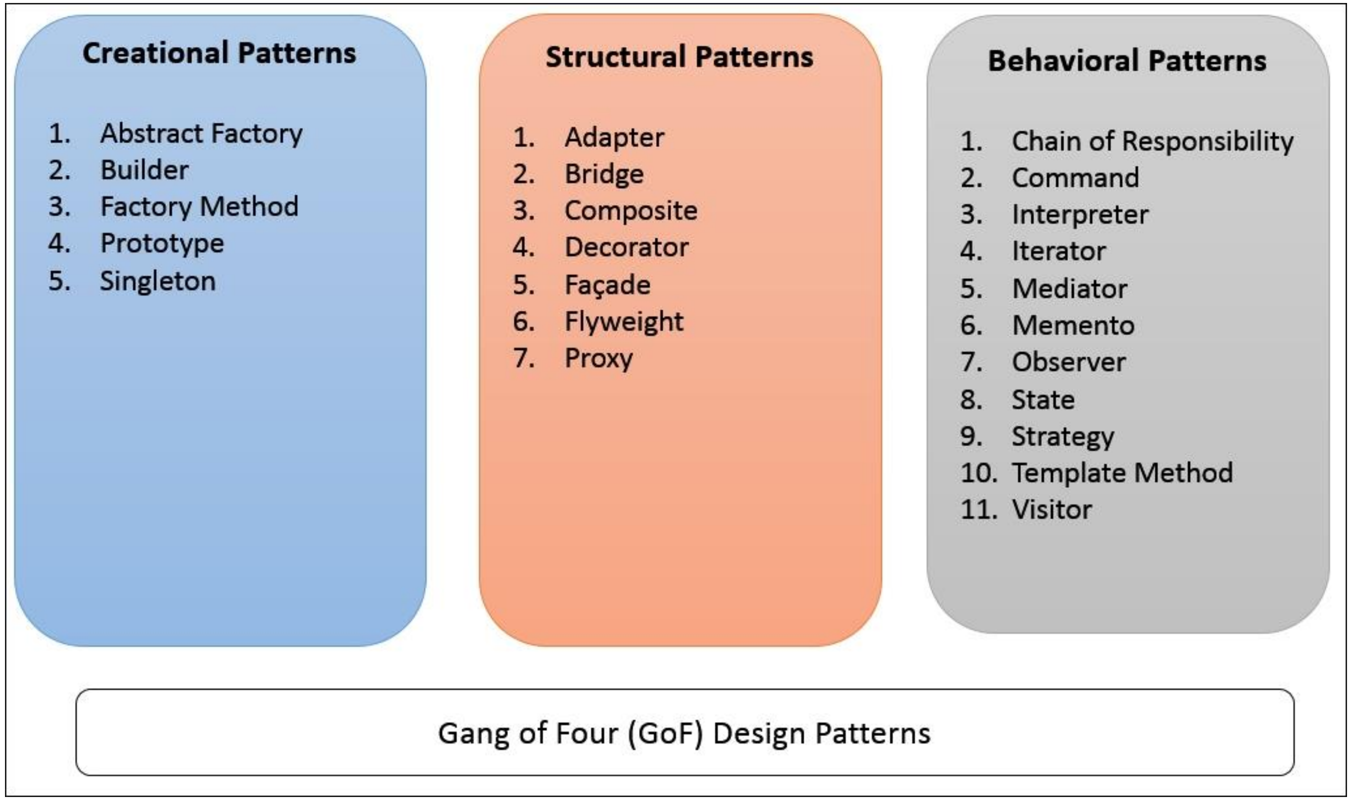
ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Introduction

- Design Patterns: Elements of Reusable Object-Oriented Software


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Introduction



Creational Patterns

1. Abstract Factory
2. Builder
3. Factory Method
4. Prototype
5. Singleton

Structural Patterns

1. Adapter
2. Bridge
3. Composite
4. Decorator
5. Façade
6. Flyweight
7. Proxy

Behavioral Patterns

1. Chain of Responsibility
2. Command
3. Interpreter
4. Iterator
5. Mediator
6. Memento
7. Observer
8. State
9. Strategy
10. Template Method
11. Visitor

Gang of Four (GoF) Design Patterns

Literature Review

- Zanoni et al. [11]
- Authors proposed a machine learning based technique
- Identify 5 specific design patterns :
- singleton, adapter, composite, decorator and factory method.

MARPLE

- The MARPLE (Metrics and Architecture Recognition PLug-in for Eclipse)
- Project focuses on the development of a complete tool for the recognition of software architectures and of design patterns
- Java programs.
- Design Pattern Clues,
- Code structures

Literature Review

- Accuracy for all patterns varies from 0.81 to 0.93
- Limitations:
- training sets used in the experiment is based on a manual design pattern.
- labeling is done using a limited (10) number of publicly available software projects.
- the contents of libraries are not included in them.
- Classifier performances are estimated under the assumption that the Joiner has 100% recall

Literature Review

- Authors[12] proposes deep learning driven approach
- Detects six design patterns:
- singleton, factory method, composite, decorator, strategy and template method.
- Nine open source repositories with design patterns as a data set.
- The accuracy of final models for each pattern varies from 0.754 till 0.985.

Literature Review

- Satoru et al. [13] proposed design patterns detection techniques using source code metrics and machine learning.
- The authors proposed approach aimed at identifying five design patterns (singleton, template method, adapter, state, strategy).
- Derived experimental data into small-scale and large-scale codes and found different set of metrics for two types of data
- For classification of the design patterns a neural network was used.
- The F-measure of proposed technique varies from 0.67 to 0.69

Literature Review

- Their proposed model is based on the semantic graphs and the pattern signature which characterize each design pattern.
- The proposed two phase approach was tested on three open source benchmark projects: JHotDraw 5.1, JRefactory 2.6.24 and Junit

Literature Review

- Addressed Scalability problems which emerged because of variants of design patterns implementation.
- The proposed approach was validated on only two major Java libraries which is not quite enough the establish the effectiveness of the authors [21] proposed method.

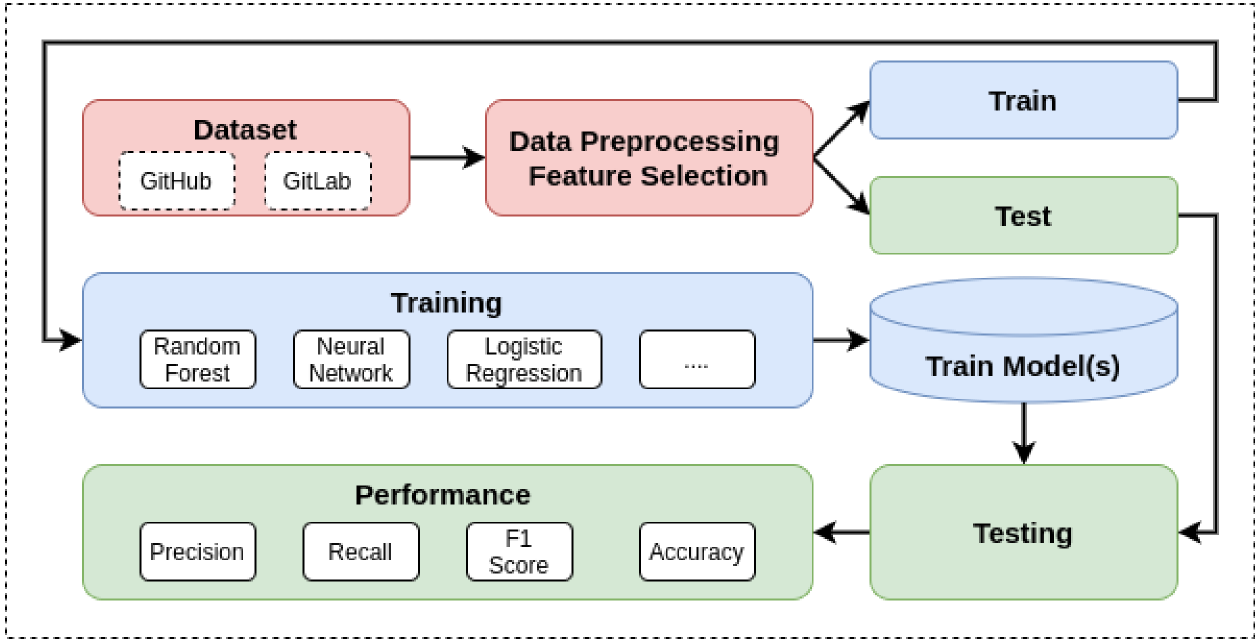
Research Gap

- lack of benchmark data for evaluation.
- The paper we present a first step in benchmark dataset creation and comparison of machine learning methods evaluated on the dataset.

Proposed Pipeline

Our proposed model pipeline contains four main stages:

- A. Data Extraction
- B. Data Prepossessing
- C. ML model definition and training
- D. Performance Measuring


Dataset
GitHub
GitLab
Data Preprocessing
Feature Selection
Train
Test
Training
Random Forest
Neural Network
Logistic Regression
....
Train Model(s)
Performance
Precision
Recall
F1 Score
Accuracy
Testing

Data Extraction

- The data set is extracted from two popular version control systems namely, GitHub1 and GitLab2 .
- In addition, to open source projects from the version control systems, projects from the university (Innopolis University) students were also added to the data set.

Data Preprocessing

- The CK metrics is used to measure some characteristics of OO systems such as classes, message passing, inheritance, and encapsulation.
- Chidamber and Kemerer (CK) metrics can "assist users in understanding object oriented design complexity and in forecasting external software qualities for example software defects, testing, and maintenance effort" [13].

CK Metrix

- Weighted Method per Class (WMC)
- Depth of Inheritance Tree (DIT)
- Number of Children (NOC)
- Coupling between Objects (CBO)
- Response for a Class (RFC)
- Lack of Cohesion in Methods (LCOM)
- Open source tool : https://github.com/mauricioaniche/ck/

Machine Learning models

- Tree Methods : Decision tree (DT) [26]
- • Ensemble Methods : Random Forest [27]
- • Gradient Boosting : Catboost [28]
- • Probabilistic Methods : Naive bayes [29]
- • Deep Learning : Artificial neural network [30]
- • Linear models : Logistic Regression [31]
- • Other : K-NN [32] & Support vector machine (SVM) [33]

default training parameters set by sklearn python library are used.

Machine Learning models

- sklearn library for training and testing the models
- Default training parameters set by sklearn python library are used.

Performance Metrix

- The five standard performance metrics are used in this paper and are namely:

Precision, recall, F1-score, weighted F1-score and accuracy.

Results till now

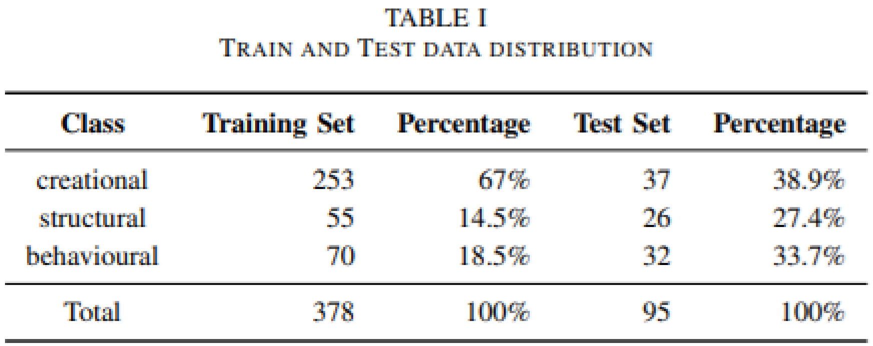


TABLE I
TRAIN AND TEST DATA DISTRIBUTION

Class	Training Set	Percentage	Test Set	Percentage
creational	253	67%	37	38.9%
structural	55	14.5%	26	27.4%
behavioural	70	18.5%	32	33.7%
Total	378	100%	95	100%

Results till now

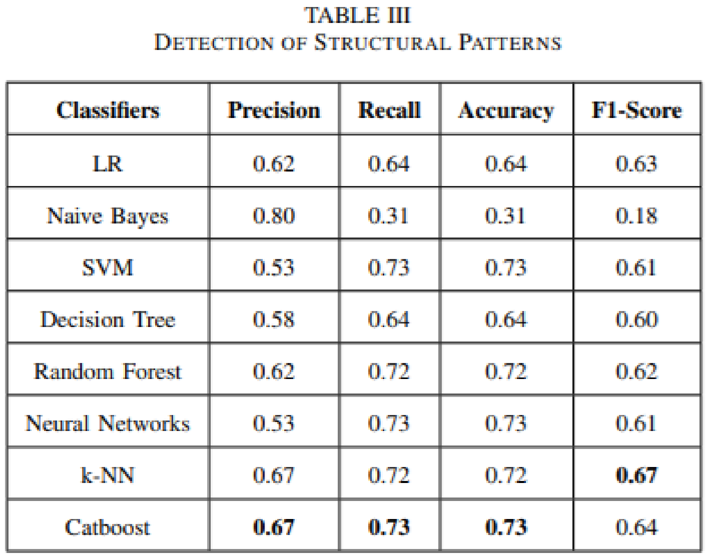


TABLE III
DETECTION OF STRUCTURAL PATTERNS

Classifiers	**Precision**	**Recall**	**Accuracy**	**F1-Score**
LR	0.62	0.64	0.64	0.63
Naive Bayes	0.80	0.31	0.31	0.18
SVM	0.53	0.73	0.73	0.61
Decision Tree	0.58	0.64	0.64	0.60
Random Forest	0.62	0.72	0.72	0.62
Neural Networks	0.53	0.73	0.73	0.61
k-NN	0.67	0.72	0.72	**0.67**
Catboost	**0.67**	**0.73**	**0.73**	0.64

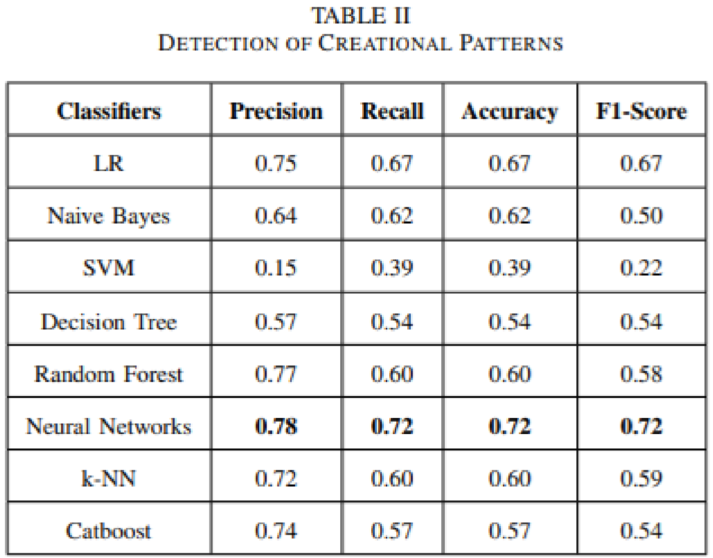


TABLE II
DETECTION OF CREATIONAL PATTERNS

Classifiers	**Precision**	**Recall**	**Accuracy**	**F1-Score**
LR	0.75	0.67	0.67	0.67
Naive Bayes	0.64	0.62	0.62	0.50
SVM	0.15	0.39	0.39	0.22
Decision Tree	0.57	0.54	0.54	0.54
Random Forest	0.77	0.60	0.60	0.58
Neural Networks	**0.78**	**0.72**	**0.72**	**0.72**
k-NN	0.72	0.60	0.60	0.59
Catboost	0.74	0.57	0.57	0.54

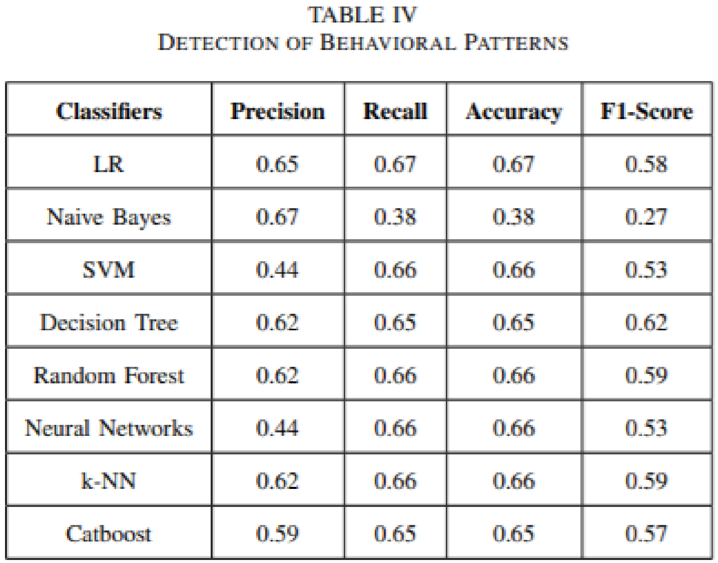


TABLE IV
DETECTION OF BEHAVIORAL PATTERNS

Classifiers	**Precision**	**Recall**	**Accuracy**	**F1-Score**
LR	0.65	0.67	0.67	0.58
Naive Bayes	0.67	0.38	0.38	0.27
SVM	0.44	0.66	0.66	0.53
Decision Tree	0.62	0.65	0.65	0.62
Random Forest	0.62	0.66	0.66	0.59
Neural Networks	0.44	0.66	0.66	0.53
k-NN	0.62	0.66	0.66	0.59
Catboost	0.59	0.65	0.65	0.57

Results till now

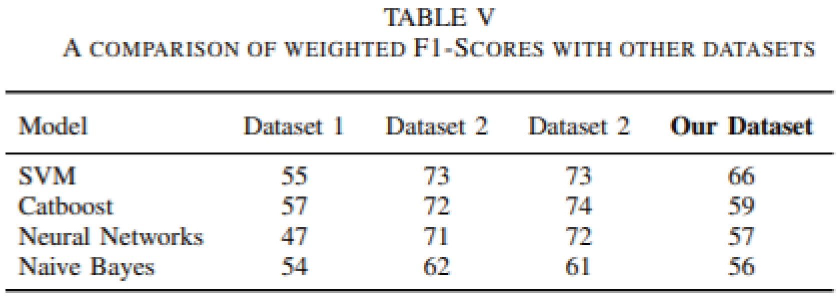


TABLE V
A COMPARISON OF WEIGHTED F1-SCORES WITH OTHER DATASETS

Model	Dataset 1	Dataset 2	Dataset 2	**Our Dataset**
SVM	55	73	73	66
Catboost	57	72	74	59
Neural Networks	47	71	72	57
Naive Bayes	54	62	61	56

Metrics

- **Precision**: Model precision score represents the model’s ability to correctly predict the positives out of all the positive predictions it made.
- **Recall**: Model recall score represents the model’s ability to correctly predict the positives out of actual positives.
- **Model accuracy** is a machine learning model performance metric that is defined as the ratio of true positives and true negatives to all positive and negative observations.
- F1-score is **harmonic mean of precision and recall score**
- **F1 Score = 2* Precision Score * Recall Score/ (Precision Score + Recall Score/)**

Thankyou