Chapter 3

METHODOLOGY AND TOOLS

3.1 Research Methodology

The study is informed by the literature on software defects, as well as literature on software quality to develop an underpinning framework for the research. The research methodology is a multi-method, multi-case, principally inductive study. A run of the mill programming deficiency forecast process incorporates two stages. Initial, an issue expectation model is fabricated utilizing past programming measurements and deficiency information having a place with every product module (class or technique level). After this preparation stage, flaw names of project modules can be assessed utilizing this model [54].

A large portion of the product deficiency forecast concentrates on creating issue indicators utilizing past deficiency information. In any case, there are situations when past flaw information is not accessible. For instance, a product organization may begin to deal with another undertaking area or might arrange building flaw indicators without precedent for their advancement cycle. What's more, current programming rendition's deficiency information won't be gathered and thusly, there won't exist in any past flaw information for the following arrival of the product. In these cases, administered learning approaches can't be created on account of the nonattendance of class marks.

For the most part, for programming grouping, records capacity, or methods of a project are considered as elements. References to variables by a capacity or calls to different modules by a capacity or its naming tradition are dealt with as the elements of that substance. A conceivable grouping produced by a bunching method is known as an allotment.
Chapter 3: Methodology and Tools

There are a couple of studies that have attempted to manufacture a deficiency forecast model when the flaw marks for modules are inaccessible.

Our proposed methodology is categorized in four parts viz. Categorization, Chi Square Test, Software Measures and Optimization.

3.1.1 Research Type

The secondary data have been used that is group of Java Programs.

3.1.2 Research Design

1. We have developed our framework in Java and Net beans environment.
2. We categorized our framework for selecting the modularity from six different choices. The six different choices are 1-10, 11-20, 21-30, 31-40, 41-50 and > 50.
3. Then for arranging object oriented parameters that are class, object, inheritance and dynamic behavior in proper groups k-means algorithm has been applied.
4. Then for testing the suitability of the programs we have applied chi-square test based on four different object oriented parameters.
5. After that Software metrics F-measure, OR and PO are used to check the strength of the qualified data.
6. Finally Particle Swarm Optimization (PSO) has been applied for finding the optimal solution of the module.
Chapter 3: Methodology and Tools

3.1.3 Tools for Data Collection

Secondary Data: Various groups of Java programs.

3.1.4 Analysis and Interpretation of Data

Then which program is qualified for the further testing and optimization we have used chi-square test. The chi-square test is a statistical tool that can be used to determine whether observed frequencies are significantly different from expected frequencies. Based on the outcome of the chi-square test we will either reject or fail to reject the null hypothesis.

The test statistic is \( \chi^2 = \sum \frac{(E-O)^2}{E} \) where E and O are the expected and observed frequencies per category.

3.2 Clustering

The bunching strategies make gatherings of articles and questions in one group are comparable while objects in various groups are unique \[28\]. Bunching is an exploration territory in information mining and particularly it situates in back-handed information mining sub bunch. Information mining is assembled into two sections as takes after:

1. **Direct data mining**: Order, estimation, forecast investigates situate here. For direct information mining, "a few variables are singled out as targets" \[55\].

2. **Indirect data mining**: Bunching, affiliation rules, representation looks into are in this examination bunch. For aberrant information mining, "no variable is singled out as an objective, and the objective is to find connections among all the variables" \[55\].
3.2.1 Clustering Analysis

The regular methodology of all the grouping system introduced here is to discover bunch focus that will speak to every group. A bunch focus is an approach to tell where the heart of every group is found, so that later when given a data vector, the framework can advise which group this vector has a place with by measuring a likeness measurement between the information vector and the whole group focus.

A percentage of the bunching procedure depends on knowing the quantity of groups. K-means and C-means clusters are of that sort. In different cases it is not important to have the quantity of bunches known from starting; rather the calculations begins by finding the principal vast group, and afterward goes an issue of known bunch numbers can be connected; notwithstanding if the quantity of bunches is not known, K-implies and fluffy C-implies bunching can't be utilized.

Cluster analysis has four basic steps \([56]\):

1) **Feature Selection**: We utilized 6 strategy level measurements including the primitive Halstead and McCabe measurements since we know the limits of these measurements.

2) **Clustering Algorithm Selection**: K- implies grouping calculation was chosen since it doesn't require the determination of bunch number, K, prior to execution of the algorithms.

3) **Cluster Validation**: Any bunching calculation can produce a few groups, yet they may not mirror the presence of the examples situating in the dataset. In this manner, assessment parameters are required to judge the adequacy of the calculation. After the bunching stage, the mean vector of every group is checked against the measurements limits
vector. Assessment parameters are utilized after this stage and they assess the general execution of our methodology.

4) **Results Interpretation**: We contrasted our model's execution and unadulterated limits based methodology. Since the execution of our two stage model enhances, we propose this methodology.

### 3.2.2 Clustering Techniques

(i) **Hierarchical Clustering**:

A progressive calculation yields a dendrogram speaking to the settled gathering of example and closeness levels at which gathering change. A dandles at which relating to the seven focuses. The dendrogram can be broken at various levels to yields distinctive bunching of the information.

Most various leveled bunching calculations are variations of the single-connection complete-interface and least fluctuation calculations. Of these the single-connection and complete connection calculations contrast in the way they describe the likeness between a couple of bunches. In the single-connection strategy, the separation between two groups is the base of the separations between all sets of examples drawn from the two bunches. In the complete-join calculations, the separation between two groups is the most extreme of all pair shrewd separation between examples in the two bunches.

(ii) **K-means Clustering**:

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the
better choice is to place them as much as possible far away from each other. Centroids change their location step by step until no more changes are done.

**Algorithm Steps**

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

**(iii) Fuzzy C-Means Clustering:**

Fuzzy c-means (FCM) is a technique for grouping which permits one bit of information to fit in with two or more bunches. This strategy (created by Dunn in 1973 and enhanced by Bezdek in 1981) is much of the time utilized as a part of example acknowledgment. It depends on minimization of the accompanying target capacity. It gives a strategy that demonstrates to gathering information focuses that populate some multidimensional space into a particular number of various bunches. Fluffy rationale tool stash order line capacity fcm begins with a beginning theory for group focuses, which are planned to check the mean area of every bunch.

**(iv) Mountain Clustering:**

The mountain grouping methodology is a straightforward approach to discover bunch fixates taking into account a thickness measure called the mountain capacity. This strategy is a straightforward approach to discover estimated
bunch focuses, and can be utilized as a preprocessor for other advanced grouping techniques.

(v) Subtractive Clustering:

The issue with the past grouping technique, mountain bunching, is that its calculation becomes exponentially with the measurement of the issue; that is on account of the mountain capacity must be assessed at every network point. Subtractive using so as to group tackles this issue information focuses as the contender for bunch focuses, rather than matrix focuses as in mountain grouping. This implies the calculation is presently relative to the issue size rather than the issue measurement.

(vi) Self-Organizing Maps:

The self-organizing map is an unsupervised neural network. This kind of system can be utilized to delineate high-dimensional information space onto a normally maybe a couple dimensional information cross section of neurons, all of which have a reference model weight vector. SOM figures out how to perceive gatherings of comparative info vectors in a manner that neurons close to one another react to comparable data vectors. The usage of SOM in genuine application ranges from discourse rearrangement of and recovery from expansive record gathering. In the last mentioned, the use of SOM has been appeared to be a decent approach. SOMs can mastermind reports with comparative substance in neighboring areas.

3.3 Clustering Issues

By Shtern \cite{4} there are a few issues which is confronting by the specialists. The vast majority of the analysts perform their work on a little arrangement of information or a product framework. The pertinence is dependably a question mark when we apply the same procedure for the enormous database. In this
manner, it is unrealistic to sum up the assessment results to other programming frameworks. The source code is no more bolstered and it is not attainable for every single other asset. Similarity is the more prominent issue. Issue discovery is additionally an issue.

We can enroll the defects in the beneath way:

• The elements that ought to be assembled together.
• The comparability measures that can be utilized to bunch together comparative elements.
• The bunching calculations to be connected.
• Evaluation of a data groups.
• There is some suited test like Chi-Square testing is required with the goal that we test the theory taking into account some descriptive properties.

3.4 Bunch Examination in Clustering

Bunch examination is a gathering of multivariate strategies whose basic role is to gathering substances taking into account their traits. They are ordered by criteria which are predefined. The target of any bunching calculation is to sort elements into gatherings, so that the variety between groups is augmented in respect to variety inside of groups.

The phases of bunch investigation procedures are

1. Certainty Extraction
2. Sifting
3. Similitude Computation
4. Bunch Creation
5. Results Visualization
6. Client Feedback Collection

Before applying bunching to a product framework, the arrangement of substances to group should be identified. Substance determination relies on upon the target of the technique. A trait is an arrangement of qualities. A quality is typically a product antique, for example, a bundle, a file, a capacity, a line of code, a database question, a bit of documentation, or an experiment. Properties might likewise be abnormal state ideas that incorporate programming ancient rarities, for example, an outline design.

Selecting a proper arrangement of traits for a given bunching assignment is critical for its prosperity Source Code Source code is the most prominent info for certainty extraction. There are two applied ways to deal with extricating truths from source code: syntactic and semantic [7]. The syntactic which is structure-construct approaches center in light of the static connections among substances. The sent out realities incorporate variable and class references, method calls, utilization of bundles, affiliation and legacy connections among classes and so on.

Some methodologies work with the data accessible in double modules. Contingent upon arrangement and linkage parameters, the double code might contain data, for example, an image table that permits proficient certainty extraction. This methodology has three points of interest:

1. It is dialect free
2. Twofold modules are the most exact and solid data, source code might have been lost or confounded to an item form of double modules. Source confuse circumstances happen as a result of human missteps, patches, moderate/unreleased renditions that are working in the generation environment and so on.
3. Module reliance data is anything but difficult to extricate from parallel modules Linkage data contains module reliance relations
Chapter 3: Methodology and Tools

The fundamental disadvantages of this methodology are that two fold meta-information data relies on upon building parameters, and that the usage of the methodology is compiler/equipment subordinate. Additionally, double code examination can't generally find all connections.

Static data is regularly deficient for recouping lost learning since it just gives restricted knowledge into the runtime way of the broke down programming; to comprehend behavioral framework properties, dynamic data is more important.

Amid the run-time of a product framework, dynamic data is gathered. The gathered data might include:

1. Object development and pulverization
2. Exceptions/mistakes
3. Method passage and way out
4. Component interface conjuring
5. Dynamic sort data
6. Dynamic part names

Execution Counters and Statistics

(a) Number of strings
(b) Size of cushions
(c) Number of Network Connections
(d) CPU and Memory Usage
Chapter 3: Methodology and Tools

(e) Number of Component Instances

(f) Average, Maximum and Minimum Response Time

There are different methods for gathering dynamic data, for example, instrumentation techniques or outsider apparatuses (debuggers, execution screens and so on). Instrumentation procedures depend on presenting new bits of code in numerous spots to identify and log every single gathered occasion. Such procedures are dialect subordinate, and not paltry to apply. After the extraction procedure is finished, a changing step might occur to guarantee that unimportant actualities are evacuated, and the accumulated truths are readied for the bunching calculation.

By there are three diverse ways to deal with powerful commotion taking care of in information examination exist planning strong calculations that are coldhearted to clamor, sifting through clamor, and amending commotion. Most vigorous calculations have a multifaceted nature control system so that the subsequent models don't over fit preparing information and sum up well to future inconspicuous information. Cross-approval, least depiction length, and auxiliary danger minimization are some normally utilized model choice standards.

So there is the need of item arranged parameter for individual beginning parameter, and afterward apply chi-square

3.5 Object Oriented Parameters

Object-arranged programming dialects have appreciated colossal fame since the 1990s, such that the item has to a great extent overwhelmed the strategy and the module as the primitive method for separating vast programming frameworks. By the 1970s, the methodology was entrenched as a method by
which substantial arrangements of orders could be broken into littler, reusable units. Parameters which can be utilized are demonstrated as follows:

3.5.1 Encapsulation

Encapsulation is the consideration inside of a project object of the considerable number of assets requirement for the item to work fundamentally, the technique and the information. The item is said to "distribute its interfaces." Other articles hold fast to these interfaces to utilize the item without being worried with how the item finishes it. The thought is "don't let me know how you do it; do what need to be done." An article can be considered as an independent iota. The item interface comprises of open strategies and instantiate information. Insurance and data stowing away are systems used to perform epitome of an article. Security is the point at which you restrict the utilization of class information or techniques. Data stowing away is the point at which you uproot information, techniques or code from a class' open interface keeping in mind the end goal to refine the extent of an item.

The classes have an open, secured and private interface. Moving techniques or information from open to ensured or to private, you are concealing the data from general society or secured interface. In the event that you have a class A with one open whole number information part d, then the definition would be

```cpp
class A
{

public:

    integer d;

};
```
In the event that you moved that information part from general society extent of the private extension, then you would be concealing the part. Better said, you are concealing the part from general society interface.

class A
{
    private:
    integer d;
};

Note that data covering up is not the same as embodiment. Because you secure or shroud techniques or information, does not mean you are exemplifying an article. In any case, the capacity to secure or conceal strategies or information, give the capacity to epitomize an article. You may say that typifying is the best possible utilization of insurance and data covering up. As a case, on the off chance that we utilized data covering up to shroud individuals that ought to obviously be in the general population interface, then we utilizing data concealing strategies, however we do not typify the class.

Figure 3.1: Object Oriented Concepts
3.5.1.1 Data Hiding

Information covering up is a normal for article arranged programming. Since an item must be connected with information in predefined classes or formats, the article can just "know" about the information it needs to think about. There is no probability that somebody keeping up the code might accidentally indicate or generally get to the wrong information inadvertently. Therefore, all information not required by an item can be said to be "covered up."

3.5.2 Data Abstraction

By typifying state, an item can guarantee that nature does not control its state in a startling way. Where a dialect backings the particular of shrouded traits, those that stay open frame an interface for the article. An article speaking to an auto might, for instance, uncover techniques that permit the driver to switch the auto on, turn left and right, change speed and to switch it off. It would not, in any case, uncover techniques that permit singular sparkle attachments to be let go such a strategy may frame part of the auto's execution, however the driver has no compelling reason to see the usage of the auto in such detail. And also by epitome, interfaces might be determined expressly for articles: objects with altogether different definitions might in any case comply with the same interface.

3.5.3 Polymorphism

Subtype polymorphism, generally called only polymorphism in the connection of article situated writing computer programs, is the capacity to make a variable, a capacity, or an item that has more than one structure. The word gets from the Greek word signifying "having various structures". On a fundamental level, polymorphism can however emerge in other figuring settings and it offers essential likenesses to the idea of decadence in science.
3.5.4 Dynamic Dispatch

Assume that vehicles have a begin Engine strategy, which turns on the vehicle's motor. Transports, autos and Lorries all keep running on inside ignition motors, so they might all utilization the same execution of begin Engine. In any case, a rocket got from the speculation of space vehicles might have a significantly more muddled startEngine strategy, which overrides that characterized for vehicles. Code that expects a vehicle and summons its begin Engine strategy might utilize a space vehicle as indicated by subtype polymorphism. In any case, the conduct of that strategy relies on upon how the conjuring is dispatched. Static dispatch chooses the conduct as indicated by the collector's static sort: it would conjure the vehicle's meaning of begin Engine, as the code was intended to work on vehicles and not space vehicles. Dynamic dispatch, be that as it may, utilizes the dynamic sort to choose conduct, so the space vehicle's form of start Engine is summoned: it decides, at run-time, which is the most proper strategy to conjure.

3.5.5 Inheritance

Inheritance is the place you have a summed up class whose conduct is stretched out in particular class.

For instance, there are distinctive shapes specifically square, rectangle, octagon and numerous others. Every one of them have regular attributes of a shape, for instance Area. You can register territory of any shape however every shape has its own particular equation for figuring zone. Such regular attributes will be caught fit as a fiddle which will then be acquired by specific classes such as square and rectangle.

There are diverse sorts of legacy. They are pictorially spoken to with constant cases in the graph appeared previously. Aside from the straightforward legacy, progressive legacy, numerous legacy and multilevel legacy, you can likewise
have a cross breed circumstance wherein you have more than one sort of legacy disordered together.

In C#, you accomplish various legacies utilizing interface. Every single other kind of legacy can be accomplished by augmenting base class in inferred class. Interface is a different component all alone. Consequently this article will cover straightforward legacy, progressive legacy and multilevel legacy. Given beneath are cases of every sort of legacy executed utilizing C#:

Example for Simple Inheritance and Hierarchical Inheritance:

class sampleShape {
    int arg1, arg2;
    public virtual int area() {
        Console.WriteLine("Virtual method area of base class");
    }
}

class sampleRectangle:sampleShape {
    public sampleRectangle(int arg1, int arg2) {
        this.arg1 = arg1;
        this.arg2 = arg2;
    }
    public override int area() {
        return (arg1 * arg2);
    }
}

class sampleTriangle:sampleShape {
    public sampleTriangle(int arg1, int arg2) {
        this.arg1 = arg1;
        this.arg2 = arg2;
    }
}
public override int area() {
    return ((arg1/2) * arg2);
}

class testClass {
    sampleShape obj = new sampleRectangle(10,20);
    int result1 = obj.area();
    obj = new sampleTriangle(10,20);
    int result2 = obj.area();
    Console.WriteLine(“Area of Rectangle:{0}, Area of Triangle:{1}”, result1, result2);
}

Here sampleRectangle gets from sampleShape, this shows straightforward legacy. Be that as it may, sampleTriangle likewise acquires from sampleShape and both the determined classes override range technique to execute the comparing recipe. This exhibits various leveled legacy.

3.5.5.1 Multilevel Inheritance

Multilevel legacy is the place base class is determined by the inferred class. What's more, the determined class is acquired further by next level of inferred class. What's more, this leveling may proceed for any level taking into account your necessity. Here is an illustration to exhibit multilevel legacy:

class baseClass {
    public virtual void displayMsg() {
        System.Console.WriteLine("Executing baseClass method...");
    }
}

class derivedClass1 : baseClass {
Chapter 3: Methodology and Tools

```csharp
class derivedClass2 : derivedClass1 {
    public void displayMsg2() {
        System.Console.WriteLine("Executing derivedClass2 method…");
    }
}

class testClass {
    public static void Main() {
        derivedClass2 obj = new derivedClass2();
        obj.displayMsg();
        obj.displayMsg1();
        obj.displayMsg2();
    }
}
```

Inferred class can get to and override techniques for every one of its guardians in the chain of command and not only its prompt base class.

### 3.6 Layers of OO Design

1. Subsystem layer contains a representation of each of the subsystems that:
   a. Enable the software to achieve its customer-defined requirements
   b. Implement the technical infrastructure that supports customer requirements.

2. Class and object layer contains:
   a. Class hierarchies that enable system to be created using generalizations
b. Representations of each object.

3. Message layer:
   a. Contains the design details that enable each object to communicate with its collaborators
   b. Establishes the external and internal interfaces for the system.

4. Responsibilities layer contains the data structure and algorithmic design for all attributes and operations for each object.

![Four Levels of Object Oriented Design](image)

**Figure 3.2 Four Levels of Object Oriented Design**

Hidden Underlying Layer → Foundation layer:

- Focuses on the design of domain objects (called design patterns).
- Domain objects provide support for human/computer interface activities, task management, and data management.
- Domain objects can be used to flesh out the design of the application itself.
- Conventional vs. OO Approaches
  - Conventional approaches: apply a distinct notation and set of heuristics to map the analysis model into a design model.
Each element of the conventional analysis model maps into one or more layers of the design model.

OOD applies:
- Data design when attributes are represented
- Interface design when a messaging model is developed
- Component-level (procedural) design for the design of operations.

3.6.1 Design Issues

Five criteria for judging a design method's ability to achieve modularity:

- Decomposability—the facility with which a design method helps the designer to decompose a large problem into sub problems that are easier to solve.
- Computability—the degree to which a design method ensures that program components (modules), once designed and built, can be reused to create other systems.
- Understandability—the ease with which a program component can be understood without reference to other information or other modules.
- Continuity—the ability to make small changes in a program and have these changes manifest themselves with corresponding changes in just one or a very few modules.
- Protection—an architectural characteristic that will reduce the propagation of side effects if an error does occur in a given module.

Five fundamental configuration rule that can be inferred for measured designs:

(1) Linguistic measured units
Chapter 3: Methodology and Tools

- Programming dialect utilized ought to be fit for supporting the measured quality characterized straightforwardly

(2) Few interfaces

- To accomplish low coupling - the quantity of interfaces between modules ought to be minimized

(3) Small interfaces (powerless coupling)

- To accomplish low coupling - the measure of data that moves over an interface ought to be minimized

(4) Explicit interfaces

- Components ought to convey in a conspicuous and direct way

(5) Information covering up

- All data around a segment is avoided outs

3.6.2 The OOD Landscape

Most important early OOD methods:

- The Booch method.
  
  - Encompasses both "micro development process" and "macro development process."

  - design context

    - macro development encompasses:

      - architectural planning activity that:

        - clusters similar objects in separate architectural partitions
Chapter 3: Methodology and Tools

- layers objects by level of abstraction
- identifies relevant scenarios
- creates a design prototype
- Validates the design prototype by applying it to usage scenarios.

- micro development:
  - defines a set of "rules" that govern:
    - the use of operations and attributes and the domain-specific policies for memory management
    - error handling and other infrastructure functions
  - develops scenarios that describe the semantics of the rules and policies
  - creates a prototype for each policy
  - instruments and refines the prototype
  - reviews each policy so that it "broadcasts its architectural vision"

- The Rumbaugh method.
  - encompasses a design activity that encourages design to be conducted at two different levels of abstraction:
    - System design -
Chapter 3: Methodology and Tools

- focuses on the layout of components that are needed to construct a complete product or system

- Analysis model is partitioned into subsystems -> allocated to processors and tasks.

- strategy for implementing data management is defined

- global resources and the control mechanisms required to access them are identified

  - Object design –

    - Emphasizes the detailed layout of an individual object.

    - Operations are selected from the analysis model and algorithms are defined for each operation.

    - Data structures appropriate to attributes and algorithms are represented.

    - Classes and class attributes are designed so they optimize data access and improve computational efficiency.

    - a messaging model is created to implement the object relationships (associations).

- The Jacobson method.

  - design model emphasizes traceability to the OOSE analysis model
3.7 Chi-Square Test

We will dependably have an invalid theory which expresses that the watched conveyance is not essentially not quite the same as the normal dissemination and obviously utilize words significant to that specific issue.

The decision rule for this test will always be $\chi^2 < \chi_c^2$ where the critical value has to be read from the $\chi^2$ distribution table. The main two numbers expected to turn upward this basic quality are the level of noteworthiness $\alpha$ furthermore, the quantity of degrees of flexibility. The degrees of opportunity for this test will be characterized as the quantity of classifications short 1. A partial $\chi^2$ distribution table is presented below. This is how we find the critical value for a particular problem: suppose that we use $\alpha = 0.05$ and have 5 degrees of freedom (6 categories). We can find the critical value by looking at the point where the two arrows in the table meet.

The test statistic is $\chi^2 = \sum \frac{(E-O)^2}{E}$ where E and O are the expected and observed frequencies per category.

Step-by-Step Procedure for Testing Your Hypothesis and Calculating Chi-Square:

1. State the theory being tried and the anticipated results. Assemble the information by leading the best possible analysis.

2. Decide the normal numbers for each observational class. Keep in mind to utilize numbers, not rates. Chi-square ought not be computed if the normal quality in any class is under 5.
3. Compute $p$ utilizing the recipe. Finish all estimations to three noteworthy digits. Round off your answer.

4. Utilize the chi-square circulation table to decide essentialness of the quality.
   - Determine degrees of opportunity and find the quality in the fitting section.
   - Locate the worth nearest to your ascertained quality on that degrees of opportunity column.
   - Move up the section to decide the $p$ esteem.

5. State your choice to the extent your hypothesis.
   - If the $p$ regard for the determined is $p > 0.5$, recognize your theory. The deviation is adequately little that hazard alone records for it.
   - If the $p$ regard for the determined is $p < 0.5$, reject your theory, and reason that some component other than chance is working for the deviation to be so inconceivable.

The chi-square test will be used to test for the "tolerability to fit" amidst watched and expected in informant.

**3.8 F- Measure (FM)**

Used to compare how similar two clustering results are. If you have a "correct" answer, then all other metrics can be compared to this to calculate relative goodness.

To begin lets say we have two clustering results:

\[ C = \text{the correct vector of bit masks} = \{c_1, c_2, c_3, \ldots c_n\} \]

\[ K = \text{the vector of bit mask results of some algorithm} = \{k_1, k_2, k_3, \ldots k_m\} \]
Then you create a “matching matrix”, $M = [a_{ij}]$. The matching matrix is just the number of cells that from result C are in cluster i, and from result K are in cluster j. So i goes from 1→n, and j goes from 1→m. N is then the total number of cells.

So imagine that the correct result C divides the data set into 3 clusters. Then we use a clustering algorithm that produces result K and divides the data into 2 clusters. The matching matrix M would be a 3 x 2 matrix beginning in the top left hand box, all the cells that were in correct cluster 1 and the clustering algorithm put into cluster 1.

The F measure is then a measure of the algorithms precision and recall.

$$F = \frac{(2 \times \text{precision} \times \text{recall})}{\text{precision} + \text{recall}}$$

**Precision (P)** = cells correctly put into a cluster / total cells put into the cluster

**Recall (R)** = cells correctly put into a cluster / All the cells that should have been in the cluster.

We can put this into an equation as:

$$P(c_{i},k_{j}) = \frac{a_{ij}}{|k_{j}|}$$

which is the number of cells in that were in both cluster i and j (from correct answer C and clustering result K, respectively) divided by the number of cells that are in cluster j.

$$R(c_{i},k_{j}) = \frac{a_{ij}}{|c_{i}|}$$

which is the number of cells in that were in both cluster i and j divided by the number of cells that are in cluster i (in this case the correct number of cells).

So then

$$F(c_{i},k_{j}) = \frac{2 \times R(c_{i},k_{j}) \times P(c_{i},k_{j})}{R(c_{i},k_{j}) + P(c_{i},k_{j})}$$

This is the F score for the comparison of one cluster to another. To calculate

$$F(C,K) = \sum_{c_{i} \text{ in } C} |c_{i}| / N \times \max_{k_{j} \text{ in } K} \{F(c_{i},k_{j})\}$$
the F score for the entire clustering result we use:

Which means, go through each cluster $i$, from all of the clusters in $C$ and find the cluster $j$ within $K$ that has the biggest F score, i.e. the cluster that was most like cluster $i$. Multiply the F score by the fraction of events in that cluster (to normalize quality to the size of the cluster). Then sum up the normalized F score for all clusters $i$ in $C$.

### 3.9 Odd Ratio (OR)

The odds ratio is a measure of effect size, describing the strength of association or non-independence between two binary data values. It is used as a descriptive statistic, and plays an important role in logistic regression.

$$OR = \frac{\text{Recall} \times (1 - \text{Precision})}{(1 - \text{Recall}) \times \text{Precision}}$$

The odds ratio is one of a range of statistics used to assess the risk of a particular outcome (software estimation) if a certain factor (or exposure) is present. The odds ratio is a relative measure of risk, telling us how much more likely it is that someone who is exposed to the factor under study will develop the outcome as compared to someone who is not exposed.

Odds are a way of presenting probabilities, but unless you know much about betting you will probably need an explanation of how odds are calculated. The odds of an event happening is the probability that the event will happen divided by the probability that the event will not happen. First, odds ratios can be calculated for case-control studies whilst relative risks are not available for such studies. Second, if we use an analysis method that corrects for confounding factors, such as logistic regression, this will report results as odds ratios. Third, odds ratios are a common way of presenting the results of a meta-analysis a statistical analysis for combining the results of several studies, used within systematic reviews.
3.10 Power (PO)

Power (PO) is defined as: $PO= ((1-Precision)^k - (1-Recall)^k)$

Finally based on the above parameters we can proceed to the comparison which is based on cost and time. We are here to improve the effectiveness of application as a software service. By the above approach we can find the better result.

3.11 Particle Swarm Optimization \[^{[57][58]}\]

Swarm Optimization is a methodology to issues whose arrangements can be spoken to as a point in a n-dimensional arrangement space. Various particles are haphazardly situated into movement through this space. At every cycle, they watch the "wellness" of themselves and their neighbors and "imitate" effective neighbors (those whose current position speaks to a superior answer for the issue than theirs) by moving towards them. Different plans for gathering particles into contending, semi-autonomous herds can be utilized, or all the particles can fit in with a solitary worldwide group. This to a great degree straightforward methodology has been shockingly viable over a mixture of issue areas.

PSO was produced by James Kennedy and Russell Eberhart in 1995 in the wake of being propelled by the investigation of flying creature rushing conduct by scholar Frank Heppner. It is identified with development enlivened critical thinking systems, for example, hereditary calculations.

3.11.1 The Algorithm

As expressed some time recently, PSO reproduces the practices of feathered creature running. Assume the accompanying situation: a gathering of feathered creatures are arbitrarily seeking nourishment in a range. There is stand out bit
Chapter 3: Methodology and Tools

of nourishment in the range being looked. All the feathered creatures don't know where the sustenance is. So what's the best method to discover the nourishment? The viable one is to take after the fowl which is closest to the nourishment.

PSO gained from the situation and utilized it to take care of the improvement issues. In PSO, each one single arrangement is a "fledgling" in the hunt space. We call it "particle". All of particles have wellness values which are assessed by the wellness capacity to be upgraded, and have speeds which administer the flying of the particles. The particles fly through the issue space by emulating the current ideal particles. PSO is introduced with a gathering of arbitrary particles (arrangements) and afterward hunt down optima by redesigning eras. In every cycle, every particle is overhauled by taking after two "best" values. The first is the best arrangement (wellness) it has accomplished as such. (The wellness quality is likewise put away.) This worth is called pbest. An alternate "best" esteem that is followed by the particle swarm analyzer is the best esteem, acquired so far by any particle in the populace. This best esteem is a worldwide best and called gbest. At the point when a particle partakes of the populace as its topological neighbors, the best esteem is a neighborhood best and is called lbest.

In the wake of discovering the two best values, the particle upgrades its speed and positions with taking after comparison (an) and (b).

\[
v[] = v[] + c1 \times \text{rand()} \times (\text{pbest}[] - \text{present}[]) + c2 \times \text{rand()} \times (\text{gbest}[] - \text{present}[]) (a)
\]

\[
\text{present}[] = \text{present}[] + v[] (b)
\]

\(v[]\) is the particle velocity, \(\text{present}[]\) is the current particle (solution). \(\text{pbest}[]\) and \(\text{gbest}[]\) are defined as stated before. \(\text{rand}()\) is a random number between (0,1). \(c1, c2\) are learning factors. usually \(c1 = c2 = 2\).

The pseudo code of the procedure is as follows
For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
   If the fitness value is better than the best fitness value (pBest) in history
   set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
   Calculate particle velocity according equation (a)
   Update particle position according equation (b)
End
While maximum iterations or minimum error criteria is not attained
Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

3.11.2 PSO parameter control

From the above case, we can learn that there are two key steps when applying PSO to optimization problems: the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. It is not like GA, which needs to change to binary encoding, or special genetic operators have to be used. For example, we try to find the solution for \( f(x) = x_1^2 + x_2^2 + x_3^2 \), the particle can be set as \((x_1, x_2, x_3)\), and fitness function is \( f(x) \). Then we can use the standard procedure to find the optimum. The searching is a repeat process, and the stop criteria are that the
maximum iteration number is reached or the minimum error condition is satisfied.

There are not many parameter need to be tuned in PSO. Here is a list of the parameters and their typical values.

The number of particles: the typical range is 20 - 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

Dimension of particles: It is determined by the problem to be optimized,

Range of particles: It is also determined by the problem to be optimized, you can specify different ranges for different dimension of particles.

Vmax: it determines the maximum change one particle can take during one iteration. Usually we set the range of the particle as the Vmax for example, the particle (x1, x2, x3) X1 belongs [-10, 10], then Vmax = 20

Learning factors: c1 and c2 usually equal to 2. However, other settings were also used in different papers. But usually c1 equals to c2 and ranges from [0,4]

The stop condition: the maximum number of iterations the PSO execute and the minimum error requirement. For example, for ANN training in previous section, we can set the minimum error requirement is one misclassified pattern. The maximum number of iterations is set to 2000. This stop condition depends on the problem to be optimized.

Global version vs. local version: we introduced two versions of PSO. Global version is faster but might converge to local optimum for some problems. Local version is a little bit slower but not easy to be trapped into local optimum. One can use global version to get quick result and use local version to refine the search.
3.12 Expected Benefits and Details of Project Objective

Examination of Data quality is a vital issue which has been tended to as information warehousing, information mining and data frameworks. It has been concurred that poor information quality will affect the nature of consequences of investigations and that it will along these lines sway on choices made on the premise of these outcomes. An endeavor to enhance grouping exactness by pre-bunching did not succeed. Be that as it may, blunder rates inside of bunches from preparing sets were emphatically corresponded with mistake rates inside of the same groups on the test sets. This marvel could maybe utilize to create certainty levels for expectations. The principle and the normal issue that the product business needs to face is the upkeep expense of modern programming frameworks. One of the primary explanations behind the high cost of support is the innate trouble of comprehension programming frameworks that are huge, mind boggling, conflicting and incorporated. The primary explanation for the above marvels is a direct result of various sizes and level of courses of action. Decaying a product framework into littler, more sensible subsystems can help the procedure of comprehension it significantly. Distinctive calculations develop diverse deteriorations. Hence, it is imperative to have techniques that assess the nature of such programmed deteriorations. In our work we introduce a brief study on programming quality forecast through grouping and after that apply chi square test for better estimation of the module.