LOVELY PROFESSIONAL UNIVERSITY

To Reduce Processing time to generate frequent itemset using Laplace equation in Apriori algorithm

A Dissertation Proposal submitted

By

Shilpi Singla

to

Department of Computer Science and Engineering

In partial fulfilment of the Requirement for the Award of the Degree of

Master of Technology in Computer Science and Engineering

Under the guidance of

Mr. Arun Malik

(May 2015)

INDIA'S LARGEST UNIVERSITY*

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of:Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the student	:Shilpi Singla	Registration No	:11002166
Batch	:2010-2015	Roll No	:RK2006A21
Session	:2014-2015	Parent Section	:K2006

Details of Supervisor:

Name	:Arun Malik	Designation	: Assistant Professor
UID	:17442	Qualification	: M.Tech
		Research Exp.	:3 years

Specialization Area:Databases (pick from list of provided specialization areas by DAA)

Proposed Topics:-

1. Apriori Algorithm in Association rule based mining.
2. Classification techniques in Data mining.
3. Clustering techniques in Data mining.

1/12
17442
Signature of supervisor

PAC Remarks: Topic 1 approved. Paper expected 11057

APPROVAL OF PAC CHAIRMAN Signature: 11041 24/11/14 Date:

*Supervision should finally encircle one topic out of three proposed topics and put up for an approval before Project Approval Committee (PAC).

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to supervisor.

CERTIFICATE

This is to certify that Shilpi Singla has completed M.Tech dissertation proposal titled "**To Reduce Processing time to generate frequent itemset using Laplace equation in Apriori algorithm**" under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma.

The dissertation proposal is fit for the submission and the partial fulfilment of the conditions for the award of M.Tech Computer Science & Engg.

Date:

Signature of Advisor

Name: Mr. Arun Malik

UID:

ABSTRACT

Data mining is a way of obtaining undetected patterns or facts from massive amount of data in a database. Data mining is also known as knowledge discovery in databases (KDD). Data mining is more in demand because it helps to reduce cost and increases the revenues. Association rule mining is a major technique in the area of data mining. Association rule mining finds frequent itemsets from a set of transactional databases. Apriori algorithm is one of the earliest algorithm of association rule mining. Apriori employs an iterative approach known as levelwise search. It follows two steps: Join step and Prune step. To recover the drawbacks of existing Apriori algorithm, we have implemented an enhanced Apriori algorithm to reduce the time and complexity of the algorithm.

ACKNOWLEDGEMENT

I am highly indebted to my mentor, **Mr. Arun Malik,** for his guidance and constant supervision as well as for providing necessary information regarding the research work and also his support in completing the report. I would also like to express my special gratitude and thanks to all the faculties and friends for giving me attention and time.

DECLARATION

I Shilpi Singla declare that the dissertation proposal entitled, **“To Reduce Processing time to generate frequent itemset using Laplace Equation in Apriori algorithm”** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date:

Investigator

Regn.No.11002166

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Data Mining: Data Mining is a way of obtaining undetected patterns or facts from bulky data in a database. It is well termed as knowledge discovery in databases (KDD). The various applications of data mining are customer retention, market analysis, production control and fraud detection. Data Mining is designed for different databases such as object-relational databases, relational databases, data warehouses and multimedia databases. Data mining methods can be categorized into classification, clustering, association rule mining, sequential pattern discovery, regression etc. Amongst these methods, association rule mining is very important which results in generating strong association rules.

1.1 Applications of Data Mining: Data mining has great approach in various fields of applications. These fields are:

- **Business Applications:** In business application, database mining is one of the popular and important applications. During mining of historical data pattern and customer profile are built on the basis of the results. It is also use in retail database to find out customer databases from the records. Using mining techniques models are used to build models for the stocks. It is also used for loan and credit applications.
- **Science Application:** It is also used in various science activities. It is used in biology, molecular science, astrology and sky objects.
- **Banking Sector:** It is also used in banking sector to predict good customers based on old customers.

1.2 Association rule generation in Data Mining:

Association rules were firstly suggested by R.Agrawal which aims at finding frequent itemsets from a set of transactional databases. The discovered relationships can be characterized in the form of association rules or set of frequent item-sets. The various algorithms in associations rule mining are Apriori, FP-Growth, Direct hashing and pruning (DHP), Apriori Tid. The algorithms vary mainly in creation of candidate item-sets and about the calculation of the supports for the frequent item-sets. It is based on support and confidence values. The representation of probability are:

Support (M->N) =P (MUN)

Confidence (M->N) =P (M|N)

The rules that meets the condition of minimum support (min_supp) and minimum confidence (min_conf) values are known as strong association rules. The itemsets which appears in the data set frequently are known as frequent itemsets. If the support value of itemsets A is ≥ min_support threshold value, then itemsets A is termed as frequent itemsets. If the support value of itemsets A is < than the min_support threshold value, then itemsets A is termed as infrequent itemsets.

The process of association rule mining is categorized into two steps which are shown in figure1.1.

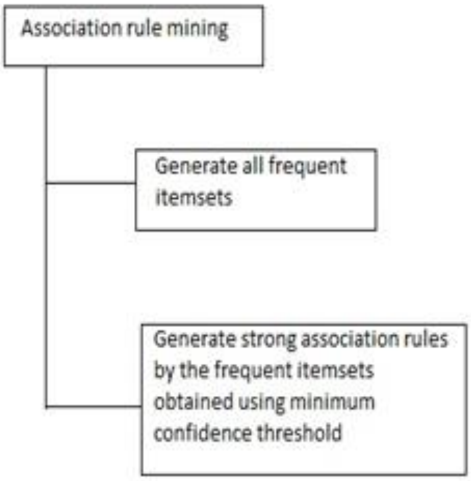

Association rule mining
Generate all frequent itemsets
Generate strong association rules by the frequent itemsets obtained using minimum confidence threshold

Figure 1.1: Association rule generation

1.3 Apriori's algorithm:

R.Agrawal and S.Srikant were the first persons to propose Apriori algorithm which is the fundamental algorithm of association rule mining presented in 1994. Apriori applies an repetitious path termed as iterative search. It follows two steps: find all itemsets that satisfy the condition of minimum support and generate strong association rules using frequent itemsets. The Apriori's algorithm is the most efficient algorithm to generate frequent itemsets.

Eg: In table 1.1 database (X) is given. The support count is 3.

Table 1.1: Example of Apriori algorithm

X

Tid	Items
K1	f1,f3,f7
K2	f2,f3,f7
K3	f1,f2,f3
K4	f2,f3
K5	f2,f3,f4,f5
K6	f2,f3
K7	f1,f2,f3,f4,f6
K8	f2,f3,f4,f6
K9	f1
K10	f1,f3

Tid	Items	SOT
K1	f1,f3,f7	3
K2	f2,f3,f7	3
K3	f1,f2,f3	3
K4	f2,f3	2
K5	f2,f3,f4,f5	4
K6	f2,f3	2
K7	f1,f2,f3,f4,f6	5
K8	f2,f3,f4,f6	4
K9	f1	1
K10	f1,f3	2

C1

Itemset	Sup_count
f1	5
f2	7
f3	9
f4	3
f5	1
f6	2
f7	2

L1

Items	Sup_count
f1	5
f2	7
f3	9
f4	3

X1

Tid	Items	SOT
K1	f1,f3	2
K2	f2,f3	2
k3	f1,f2,f3	3
k4	f2,f3	2
k5	f2,f3,f4	4
k6	f2,f3	2
k7	f1,f2,f3,f4	4
k8	f2,f3,f4	3
k10	f1,f3	2

C2

Itemset	Sup_count
f1,f2	2
f1,f3	4
f1,f4	1
f2,f3	7
f2,f4	3
f3,f4	3

L2

Item set	Sup_count
f1,f3	4
f2,f3	7
f2,f4	3
f3,f4	3

X2

Tid	Items	SOT
K3	f1,f2,f3	3
K5	f2,f3,f4	3
K7	f1,f2,f3,f4	4
K8	f2,f3,f4	3

Itemset	Sup_count
f2,f3,f4	3

C3

L3

Itemset	Sup_count
f2,f3,f4	3

The steps(S) of the algorithm is as follows:

S1: At first, transform the given database X into meaningful database i.e with Size_of_transaction column.

S2: Then, place items in an iterative way and give the frequency of items for finding the candidate generation rule.

S3: The number of items produced in each transactions are termed as size_of_transaction.

S4: L1 can be generated on the basis of min_support.

S5: Support count of v5, v6, v7 < 3, hence ignore these data from X. When L1 is established, the size of k is 2, ignore those records having Size_of_transaction 1 in X.

S6: Use join operation to generate a candidate item set i.e L1 ∞L1.

S7: The 2-frequent itemset is generated i.e L2.

S8: When L2 is created properly, we can determine the record of 5 transactions (K1, K2,K4, K6, K10)are only two in X1..

S9: Use join operation to generate a candidate 3-item set i.e L2 ∞L2.

S10: Take C3 to create L3.

S11: Since, L3 has only 3 itemsets, therefore C4 =ϕ. The algorithm terminates and generate all frequent itemsets.

S12: The process will be continued for (C_K) until C_k+1 becomes null.

1.4 Advantages of Apriori algorithm:

- Apriori algorithm is easy to understand.
- It is simple to implement.
- It uses large itemset property.
- It is easily parallelized.

1.5 Disadvantages of Apriori algorithm:

- It requires many database scans.

- It is less efficient and accurate.
- It consumes more memory.

Chapter2

REVIEW OF LITERATURE

In this paper, **(Chanchal Yadav, 2013),** discussed about the drawbacks of classical Apriori algorithm. The various approaches which were used to overcome the limitations of classical Apriori algorithm to improve efficiency are also discussed. They also proposed a new algorithm which decreases the pruning operation of candidate itemset. The new algorithm compares the dataset according to time and rules generated and found to be an efficient algorithm for large datasets. It reduces storage space, improves efficiency and accuracy of the algorithm.

(Gang FANG, 2010), have proposed an algorithm of mining spatial topology association rule based on Apriori. This algorithm uses a spatial topological relation to mine the association rule. It follows three steps: First of all, it decreases storage space while developing mining database. Secondly, it fastly calculates the candidate support count. Thirdly, it is fast to connect new candidate itemset of previous frequent itemsets as bottom-up search technique. From the experimental result, it was shown that this algorithm is able to take out spatial multilayer transverse association rules from spatial database and is very efficient for mining short frequent topological itemsets.

(Garg, 2013), done the research on finding association rule using Apriori algorithm. In this paper, we have three association rule algorithms: Apriori Association Rule, PredictiveApriori Association Rule and Tertius Association Rule. By comparing the result of these three algorithms on the basis of parameter elapsed time using a data mining tool Weka, it was shown that Apriori Association algorithm is faster than the PredictiveApriori Association Rule and Tertius Association Rule algorithms. In future, we can combine these three algorithms for generating more efficient algorithm.

In this paper, on the basis of analysis and study of previous efforts that researcher have applied, an improved Apriori algorithm (IAA) for association rule mining is proposed by **(Huan Wu, 2009)**. The IAA conquer the limitations of original apriori algorithm. The IAA introduces a new count based approach which is used to elicit the redundant candidate itemsets and uses generation record to reduce the scanning time of database. The IAA meets the various challenges correlates with association rule mining such as reducing I/O cost, improving efficiency and increasing processing speed. From the experimental results, it was proved that IAA is better than original Apriori algorithm because IAA counts each candidate itemsets once. But C-R problem exists in IAA which could not be solved where C represents condition item sets and R represents result item sets.

(Jaishree Singh, 2013) have introduced a modified Apriori Algorithm called an improved Apriori Algorithm(IAA) to conquer the limitations of classical Apriori Algorithm. The classical Apriori algorithm scans the database many times. If database contains ample number of records, it takes huge time to scan the database which results in increasing I/O cost. The improved Apriori Algorithm reduces the scanning time by eliminating the transactions containing irrelevant records. It uses the concept of attribute named as Size_Of_Transaction (SOT) which contains the number of items exists in specific transaction. It also decreases the I/O cost. By comparing improved Apriori Algorithm with classical apriori algorithm, it was shown that improved Apriori Algorithm is better on the basis of efficiency and optimization. This algorithm has certain drawback also as it has to deal with new database after every generation of frequent itemset.

(Kasamsan, 2010), have discussed the limitations of classical Apriori algorithm. The classical Apriori algorithm takes more time to read the data from database. The new algorithm known as Boolean Algebra Compress technique for Association rule Mining (B-Compress) is proposed to recover the limitations of classical Apriori algorithm.This algorithm performs three steps: First, it will collapse the data. Secondly, it will decrease amount of time to scan database. At last, it will decrease the file size. From the result, it was shown that B-Compress technique has higher efficiency than existing Apriori rule. In future, the performance can further be improved by decreasing the number of candidates as well as produce the candidates in RAM instead of hard disks.

(P.Uma, 2012), have proposed an improved algorithm to generate frequent itemsets from a database based on existing Apriori Algorithm. The main benefit of this approach is that the database is stored in transposition form and in each repetition database is sanitized and reduced by producing a transaction id for each pattern. This method saves the time and also reduces the database size. It has been presented the experimental results, using synthetic data, showing that the improved algorithm runs faster than existing Apriori Algorithm. So, the proposed algorithm is suitable for massive datasets.

(Paul, 2010) have developed a new method distributed apriori association rule to recover the limitations of classical apriori algorithm. The new method follows the concept of knowledge elicitation. The implementation of both algorithms is shown using theory of grid computing. Grid computing is a form of distributed computing that enables the developers to work together on a single task at same time. Grid computing has capability to increase the efficiency and decreases the cost of computing networks by optimizing the resources. It is best for large workloads. By comparing, it was shown that distributed apriori association rule on grid based environment is better than classical apriori algorithm. The future scope is that knowledge could be extracted in parallel to produce more optimized result.

(R.Santhi, 2011), have proposed an effective hash based Apriori algorithm for candidate set generation. The number of candidate 2-itemsets generated by the proposed algorithm is in order of magnitude and smaller than generated by existing Apriori algorithm. Hence, it resolves the performance bottleneck. Hash based Apriori scans the database once. On comparing existing Apriori algorithm with proposed Apriori algorithm, it was shown that the proposed Apriori algorithm always outperform the existing Apriori algorithm.

In this paper, **(Rina Raval, 2013)** discussed about classical Apriori algorithm. The limitations of Apriori algorithm are also discussed. Various authors have done a survey on many good improved techniques of Apriori algorithm and concluded that many implementatins are still required basically on pruning the itemsets in Apriori to increase accuracy of algorithm.

(S.Bagga, 2014) have introduced a new method to implement Apriori algorithm in MATLAB. To implement Apriori algorithm, Attribute Affinity Matrix is used to enhance the efficiency of existing Apriori algorithm which was implemented in C. This method simply discovers the frequent data itemsets from huge amount of data and reduces the execution time and is more efficient than existing Apriori algorithm. From the results, it was shown that the Apriori algorithm implemented in MATLAB is 80% faster than Apriori Algorithm in C.

(S.Vijayarani, 2010), involves the problem of developing privacy preserving algorithms for association rule mining. Privacy has become an important factor in data mining. Data quality is also important so that the information received must be correct. We can use another data mining techniques such as classification, clustering for securing data as well as knowledge. In future we will give privacy preserving data mining based on association rule using Tabu search optimization technique.

To weed out obstructions of frequent itemsets mining in Apriori algorithm **(Wanjun Yu, 2008)** has given a new algorithm named as Reduced Apriori Algorithm with Tag (RAAT). In Apriori algorithm, there is large number of candidate 2-itemsets and less tendency to determine support value. So, it takes lot of time to scan the database repeatedly and decreases the efficiency. To improve this, RAAT uses Apriori-gen operation to form candidate 2-itemsets which results in diminishing the pruning operation. RAAT also follows the concept of tag to increase the speed of support calculation. As a result, RAAT shortens the time and improves efficiency. The experimental results shown that RAAT performs well when it is compared with Apriori algorithm in a number of times.

The classical Apriori algorithm generally focuses on only two aspects: minimum support and minimum confidence to generate strong association rule. There may be a chance that sometimes it is necessary to determine strong association rules for making decisions and sometimes less strong rules are required. To fulfill this condition, **(Wei-Min Ma, 2008)**, proposed two updated algorithms based on Apriori: AMS (Algorithm for mining stronger association rules) and AMLS (Algorithm for mining less strong association rules) which focus on three aspects: minimum support, minimum confidence and minimum interest. These algorithms works in the form of matrix to decrease the scanning time of database. On the basis of comparison of classical Apriori algorithm with AMS and AMLS, it was proved that AMS and AMLS are better than classical Apriori algorithm.

(Xue-Gang Hu, 2004) introduced a new Quantitative extended concept lattice (QECL) method which is based on the concept of lattice to mine association rule .Concept lattice is type of induced lattice which comes into being based on the partial ordering relation between O, D, R where O represents set of objects, D represents set of attributes and R is the binary relationship between O and D. This method is better than existing Apriori algorithm because we can mine association rules easily without finding frequent itemsets and easily obtained strong association rules within a reasonable amount of time. It eliminates the burden of scanning database repeatedly. As a result, the efficiency and accuracy of mining association rules are improved.

In this paper, the classical Apriori algorithm is discussed. The faults that are present in classical Apriori algorithm such as consuming more time to generate candidate itemsets, scanning the database repeatedly are also discussed. In order to remove all these faults, **(Yanfei Zhou, 2010)**, described an improved Apriori algorithm. This improved Apriori algorithm consists of three segments: First is decreasing number of judgements during the time of generating frequent candidate itemsets. Secondly, pruning frequent itemsets. Finally, optimize the database. The improved Apriori algorithm was compared with classical Apriori algorithm on the basis on different support degree, different number of trading services and different number of items. From this comparison, it was proved that improved Apriori algorithm improves performance, increases efficiency, and reduces the redundant operation while producing frequent itemsets and strong association rules.

(Yang, 2012) proposed a theorem to upgrade the traditional Apriori Algorithm. The traditional Apriori Algorithm takes more time to scan the database in order to find out the frequent itemsets. This increases the complexity and decreases efficiency. The proposed algorithm decreases the database access on the basis of customer habits. It uses relative theorems to find frequent itemsets. For applying improved Apriori algorithm to E-commerce, there will be a need to develop a shopping site because when customers visit the shopping site the system will automatically find out their next purchasing goods based on goods already available in their shopping basket. So, it will save time and increases the efficiency and provides more benefit. The customers could easily generate association rules with the help of improved Apriori algorithm and suggests useful products to customers within a reasonable time. According to experimental results, it was shown that improved Apriori Algorithm when compared with traditional Apriori algorithm is more efficient

Chapter3

PRESENT WORK

3.1 Problem Formulation

The Apriori's is the efficient algorithm to generate frequent itemsets for data mining. In the previous times, various enhancements had been proposed to upgrade its performance. In Apriori algorithm, there is ample number of candidate 2-itemsets and less tendency to determine support value. So, it takes lot of time to scan the database repeatedly and decreases the efficiency. To improve this, RAAT uses Apriori-gen operation to form candidate 2-itemsets which results in diminishing the pruning operation. As a result, RAAT shortens the time and improves efficiency. The classical Apriori algorithm scans the database repeatedly. If database contains ample number of records, it takes huge time to scan the database which results in increasing I/O cost. The improved Apriori algorithm shortens the scanning time by eliminating the transactions containing irrelevant records. It uses the concept of attribute named as Size_Of_Transaction (SOT) which contains the number of items exists in specific transaction. It also decreases I/O cost. The other enhancement in Apriori's algorithm is based on customer habits. It uses relative theorems to find frequent itemsets. For applying improved Apriori algorithm to E-commerce, there will be a need to develop a shopping site. So, it will save time and increases the efficiency and provides more benefit. The customers could easily generate association rules with the help of improved Apriori algorithm and suggests useful products to customers within a reasonable time. We worked on to decrease the number of transactions of Apriori algorithm by using Laplace equation to generate frequent itemsets.

3.2 Objectives Of Research

1. To study and analyse various association rule generation algorithms for data mining and to identify the problem of time and complexity in Apriori's algorithm.
2. To propose enhancement in the Apriori's algorithm by taking market basket dataset for frequent itemset generation in data mining.
3. To implement existing and proposed algorithm and analyze the performance of algorithms in terms of time and complexity.

3.3 Research Methodology

The Apriori algorithm is generally applied on the data set of super market to generate the frequent itemsets. The frequent itemsets are generated on the super market dataset are on the basis of number of transactions. If the data set is huge, then it takes more time to scan the database. This approach will consume time and system resources. The Apriori algorithm efficiency can be enhanced if the number of transactions will be reduced. The number of transactions, if reduce then the time required to generate the frequent itemsets will also be less. The drawback of Apriori algorithm is diminished in the proposed algorithm. In our proposed algorithm, Laplace equation is used to reduce the size of transactions and to generate frequent itemsets. Laplace Matrix defined starting point for a representation of matrix that accurately captures the inside-out transformation from graphs to geometric space is the Laplacian matrix of the graph. Given pair wise affinities, this matrix is:

$$L = D - A$$

That is L is the matrix whose diagonal contains the degrees of each object, and whose off-diagonal entries are the negated values of the adjacency matrix. Laplacian matrix is the product of the incidence matrix with its transpose, so the Laplacian can be consider as a correlation matrix for the objects, but with correlation based only on connection. The correlation matrix will be smaller than the base matrix. L is symmetric and positive semi-definite, so it has n real-valued eigen values. The smallest eigen value is 0 and the corresponding eigenvector is the vector of all 1's. The number of eigen values that are 0's corresponds to the number of connected components in the graph. In a data-mining setting, there are some points about the number of clusters present in the data, although connected components of the graph are easy clusters.

The second smallest eigen value-eigenvector pair is the most interesting in the eigen decomposition of the Laplacian, if it is non-zero, that is the graph is connected. The second eigenvector maps the objects, the nodes of the graph, to the real line. Any value in the range of the mapping, and consider only those objects mapped to a greater value, then those objects are connected in the graph. In other words, this mapping arranges the objects along a line in a way that corresponds to sweeping across the graph from one `end' to the other.

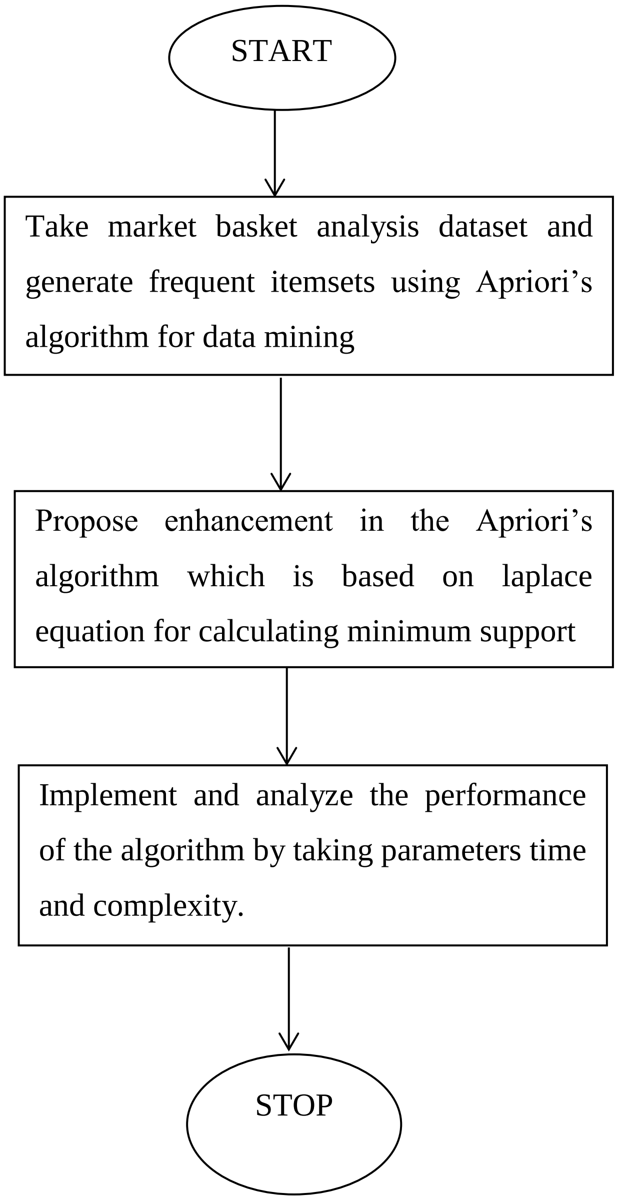

START
Take market basket analysis dataset and generate frequent itemsets using Apriori's algorithm for data mining
Propose enhancement in the Apriori's algorithm which is based on laplace equation for calculating minimum support
Implement and analyze the performance of the algorithm by taking parameters time and complexity.
STOP

Figure 3.1: Flowchart

The proposed idea is implemented in MATLAB (Matrix Laboratory). MATLAB is one of the simplest programming language that deals with mathematics. It is easy to generate frequent itemsets in MATLAB. It is easy to understand and implement. We can analyze and develop new algorithms easily.

The Pseudo Code of Enhanced Apriori algorithm is as follows:

```
Step 1. Ci: Candidate itemset of size i
Step 2. Li : frequent itemset of size i
Step 3. L=(L1(1)L2(2)L3(3)-----Ln)/C(1,2,3…..N)
Step 4. for (i = 1; Li !=Ǿ; i ++) do begin
```

```
            Ci+1 = candidates developed from Li;
Step 5. for each transaction t in database do
           increment the count of all candidates in
           Ci+1 that are contained in t
Step 6. Li+1 = candidates in Ci+1 with min_support
Step 7. End
Step 8. return ÛiLi;
```

Chapter 4

RESULTS AND DISCUSSIONS

We have implemented the proposed algorithm on windows 7 operating system, processor 2.13 Ghz, RAM 2.00 Gb. We have used MATLAB tool which is a data mining tool to make GUI interfaces. The results for both existing and enhanced Apriori algorithm are shown below:

4.1 Experimental Work

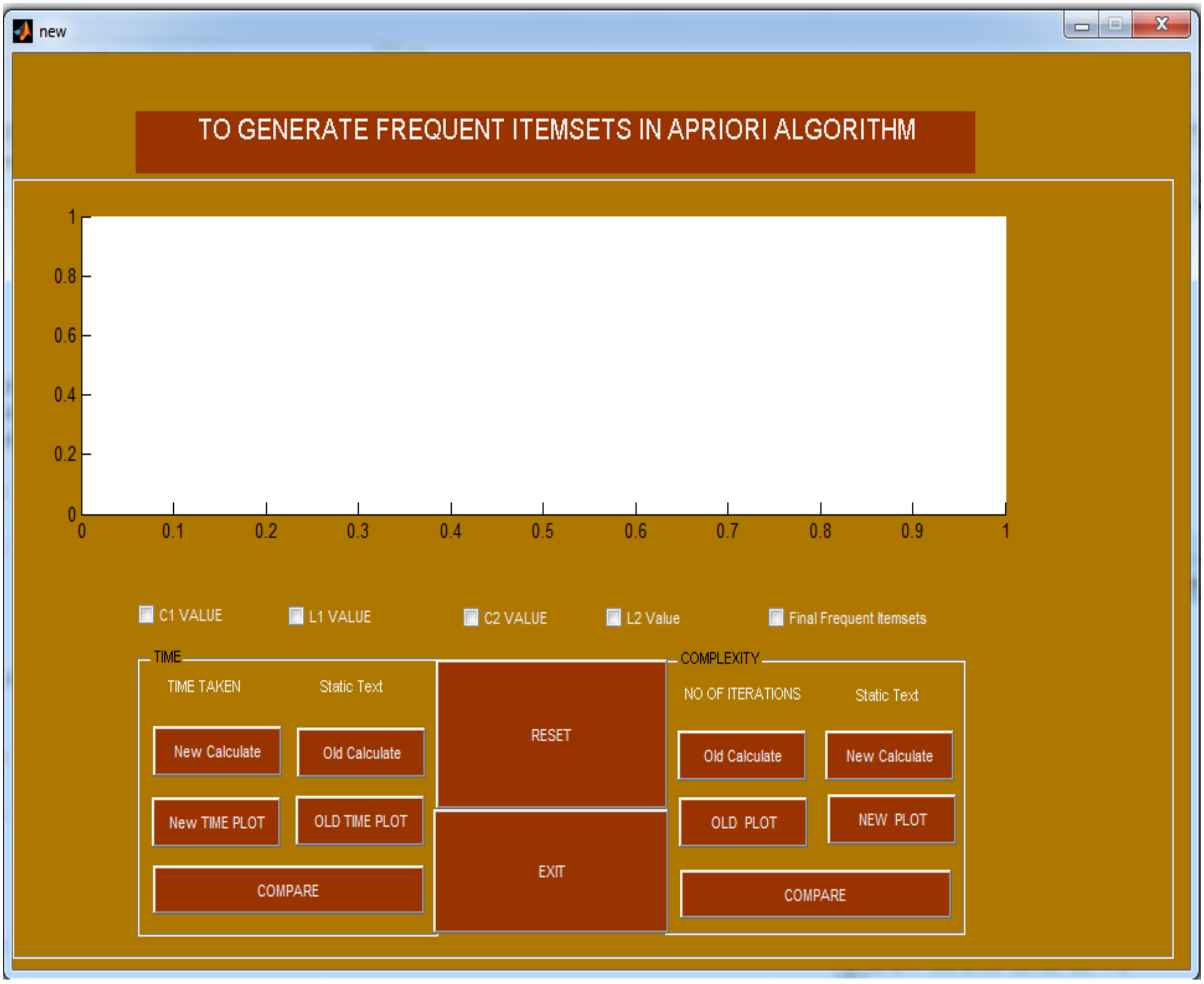
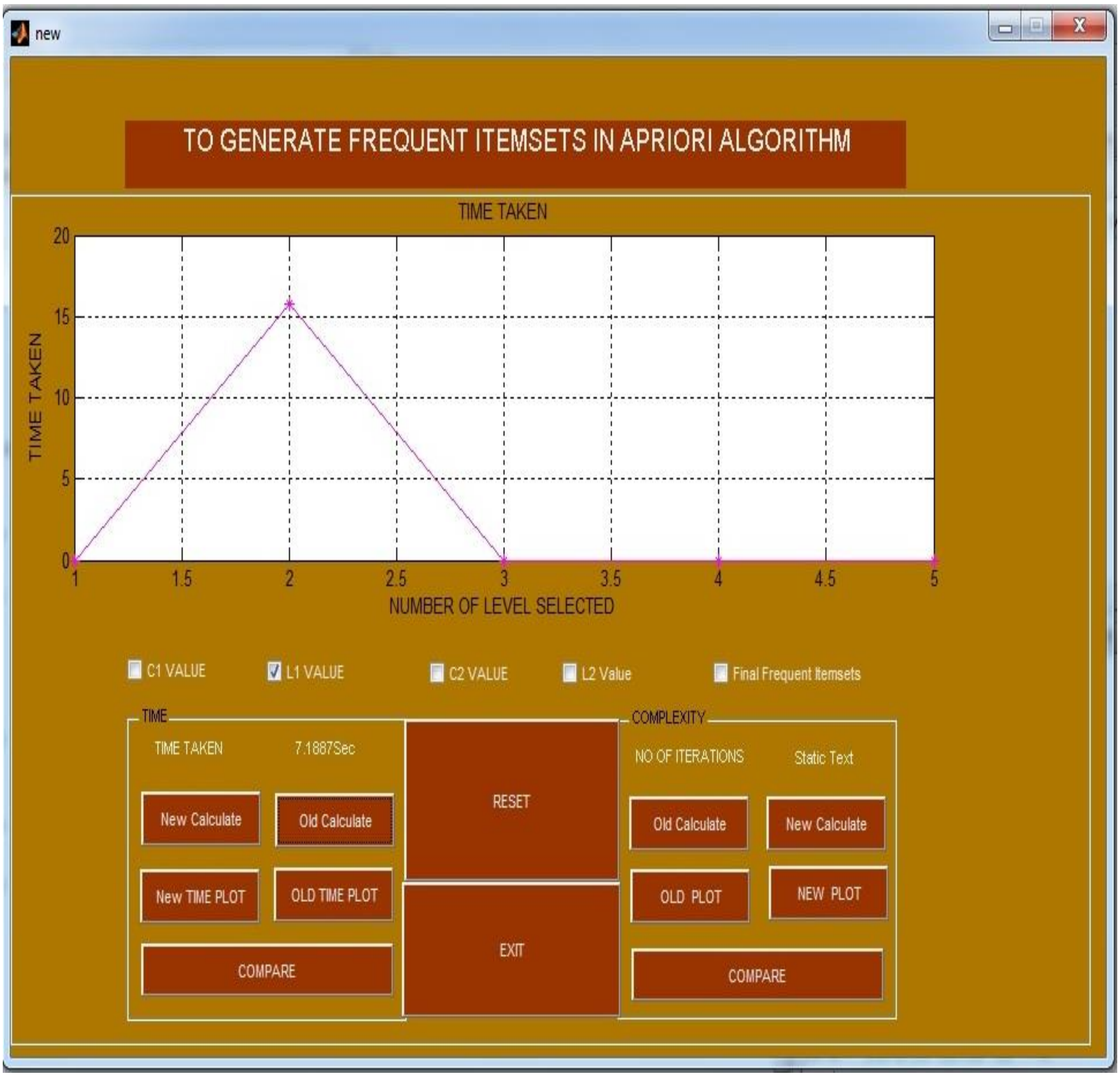


Figure 4.1: Default mode of the tool

As shown in figure 4.1, the tool is developed for frequent itemset generation in Apriori algorithm. In the tool C1, L1, C2, L2 and final frequent itemsets generation steps are defined. The time and complexity are calculated using existing and enhanced algorithms.



TO GENERATE FREQUENT ITEMSETS IN APRIORI ALGORITHM

□ C1 VALUE ☑ L1 VALUE □ C2 VALUE □ L2 Value □ Final Frequent Itemsets

TIME TAKEN 7.1887Sec

NO OF ITERATIONS Static Text

Figure 4.2: Time analysis of existing algorithm for L1 value

As illustrated in figure 4.2, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate L1 value. The existing algorithm takes 7.18 seconds to generate L1 frequent itemset.

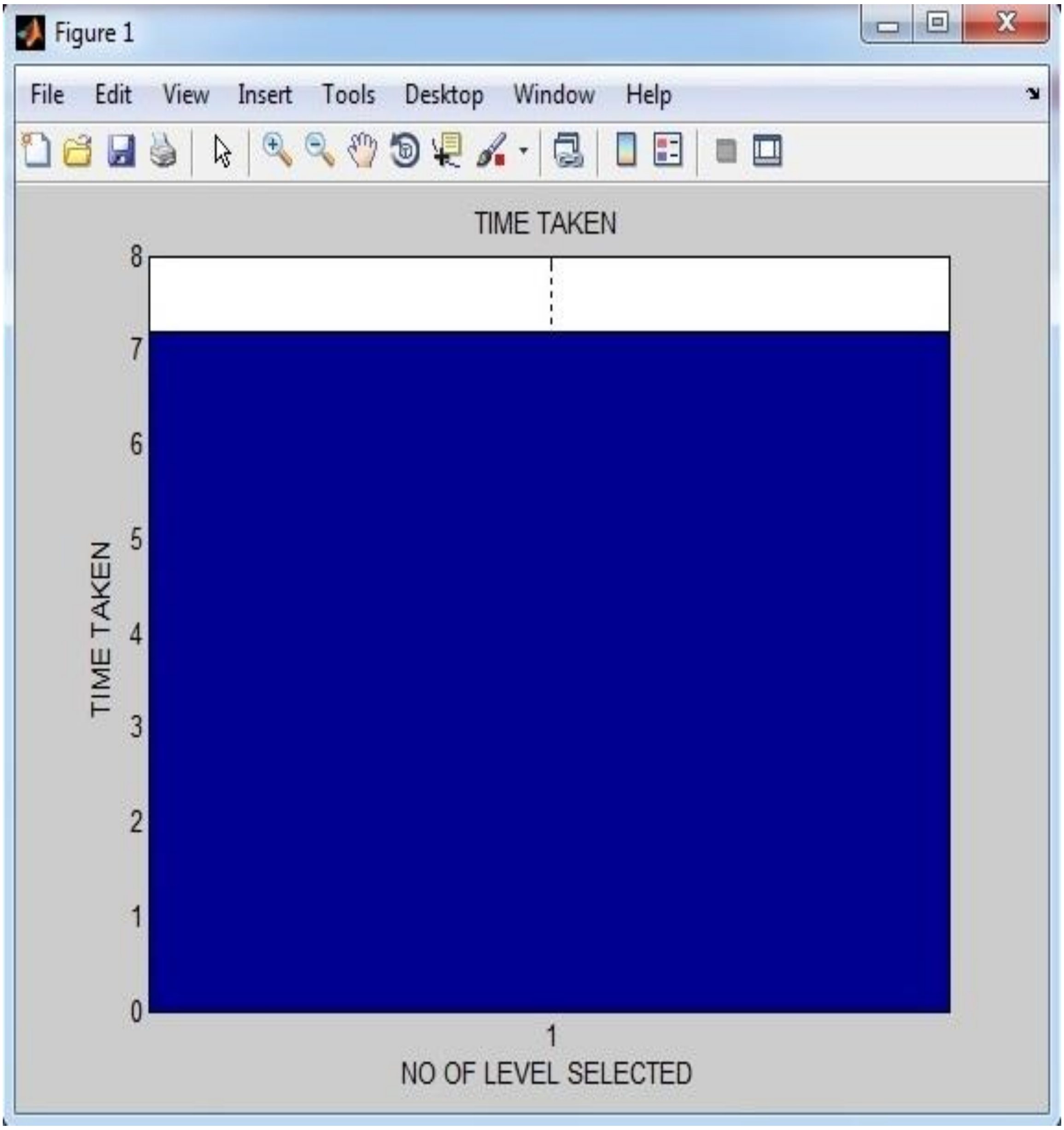
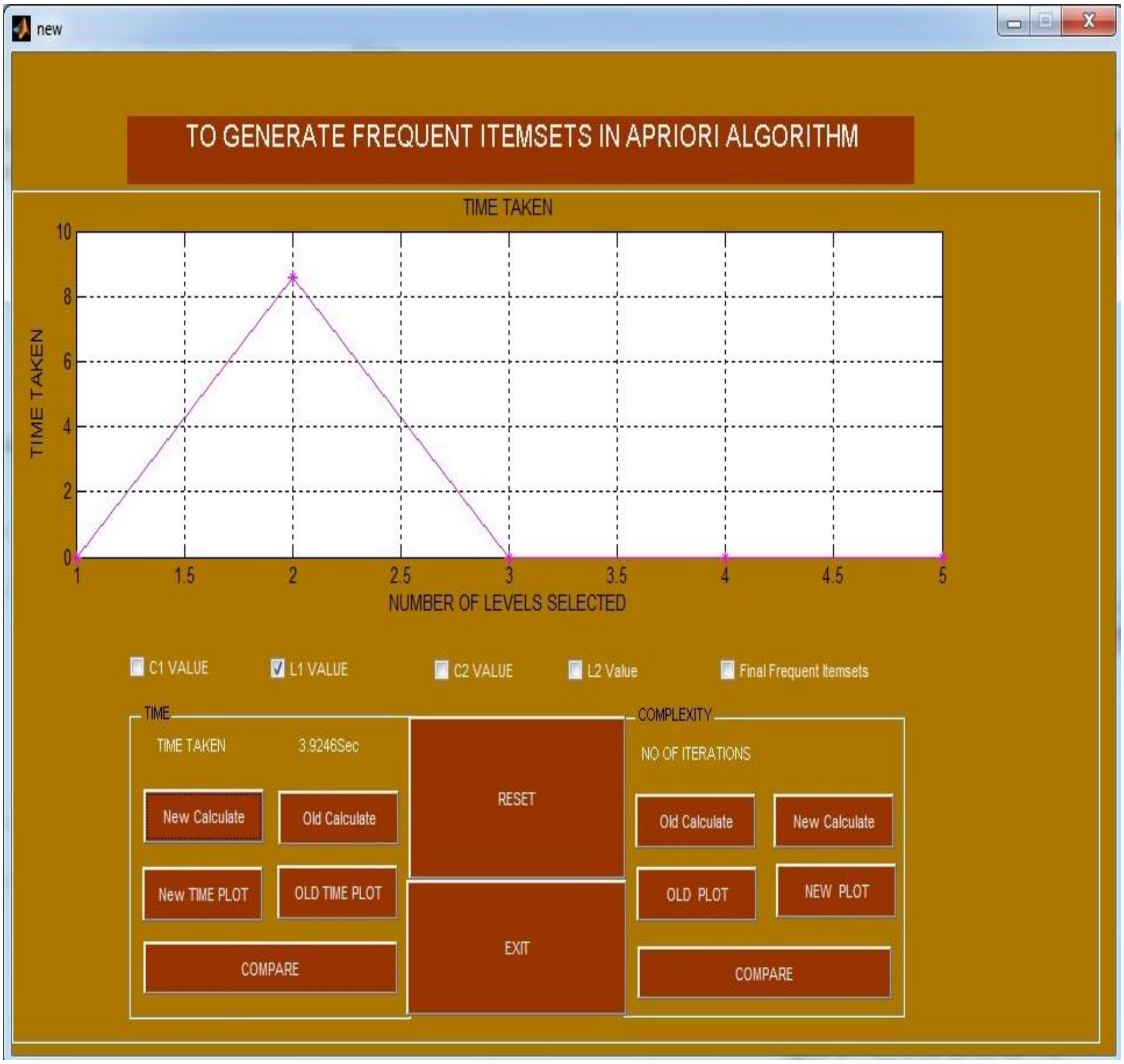


Figure 4.3: Plot of time analysis of existing algorithm for L1 value

As shown in figure 4.3, the performance analysis of Apriori’s algorithm is evaluated in terms of time taken to generate L1 value. The existing algorithm takes 7.18 seconds to generate L1 frequent itemset.



TO GENERATE FREQUENT ITEMSETS IN APRIORI ALGORITHM

☐ C1 VALUE ☑ L1 VALUE ☐ C2 VALUE ☐ L2 Value ☐ Final Frequent Itemsets

TIME TAKEN 3.9246Sec

Figure 4.4: Time analysis of enhanced algorithm for L1 value

As illustrated in figure 4.4, the performance analysis of Apriori’s algorithm is evaluated in terms of time taken to generate L1 value. The enhanced algorithm takes 3.92 seconds to generate L1 frequent itemset.

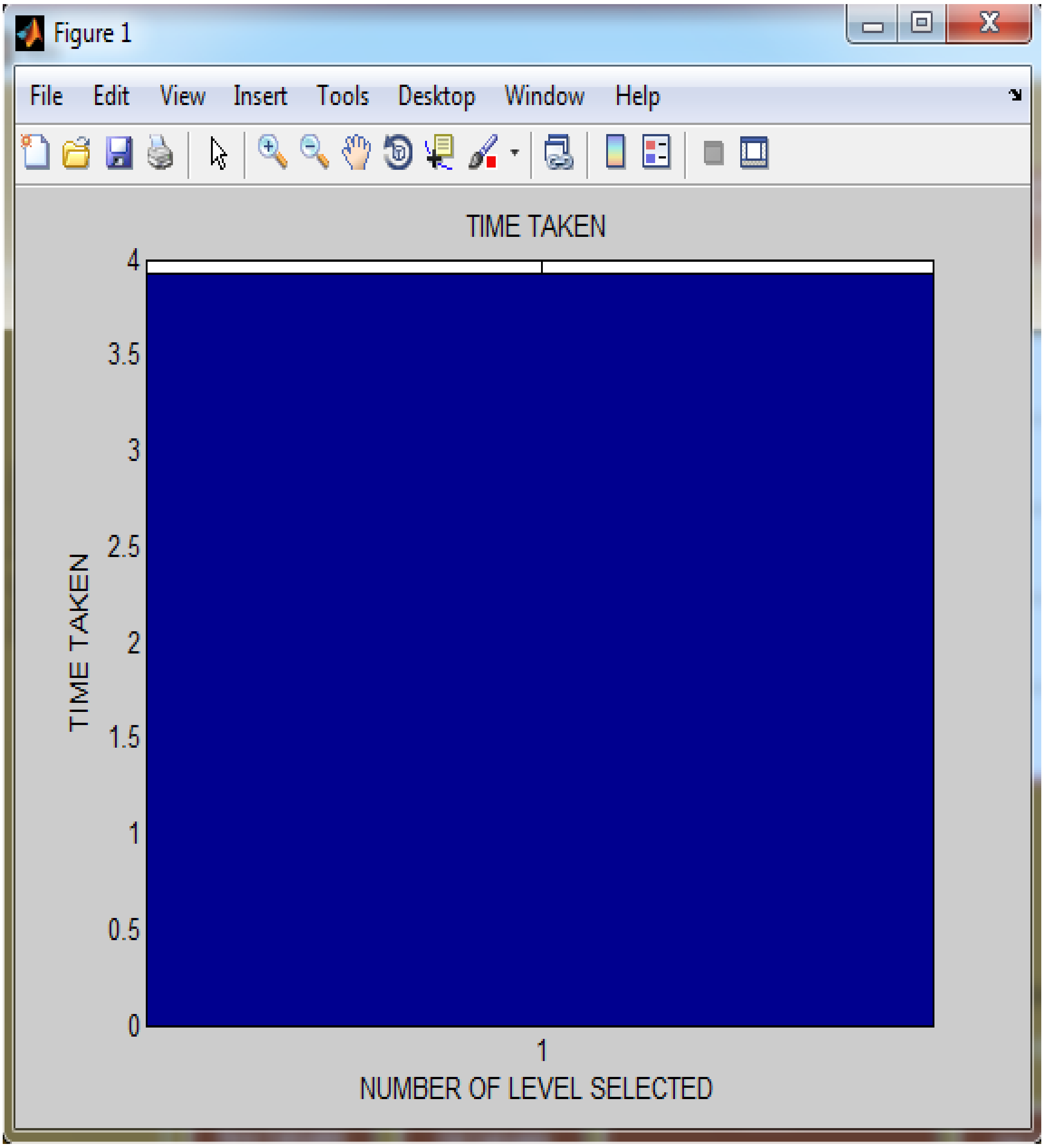
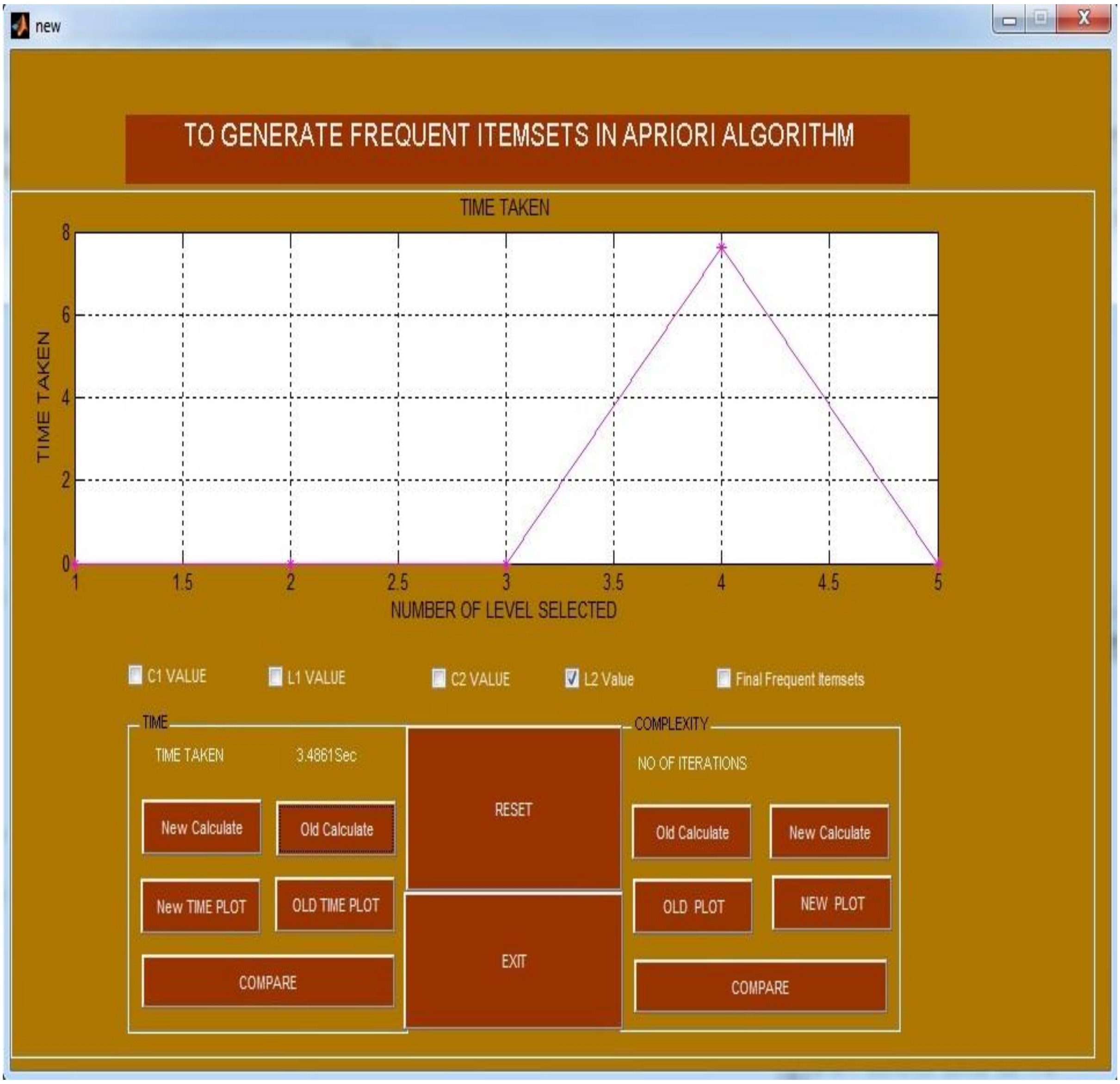


Figure 4.5: Plot of time analysis of enhanced algorithm for L1 value

As shown in figure 4.5, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate L1 value. The enhanced algorithm takes 3.92 seconds to generate L1 frequent itemset.



□ C1 VALUE □ L1 VALUE □ C2 VALUE ☑ L2 Value □ Final Frequent Itemsets

TIME TAKEN 3.4861Sec

Figure 4.6: Time analysis of existing algorithm for L2 value

As illustrated in figure 4.6, the performance analysis of Apriori’s algorithm is evaluated in terms of time taken to generate L2 value. The existing algorithm takes 3.48 seconds to generate L2 frequent itemset.

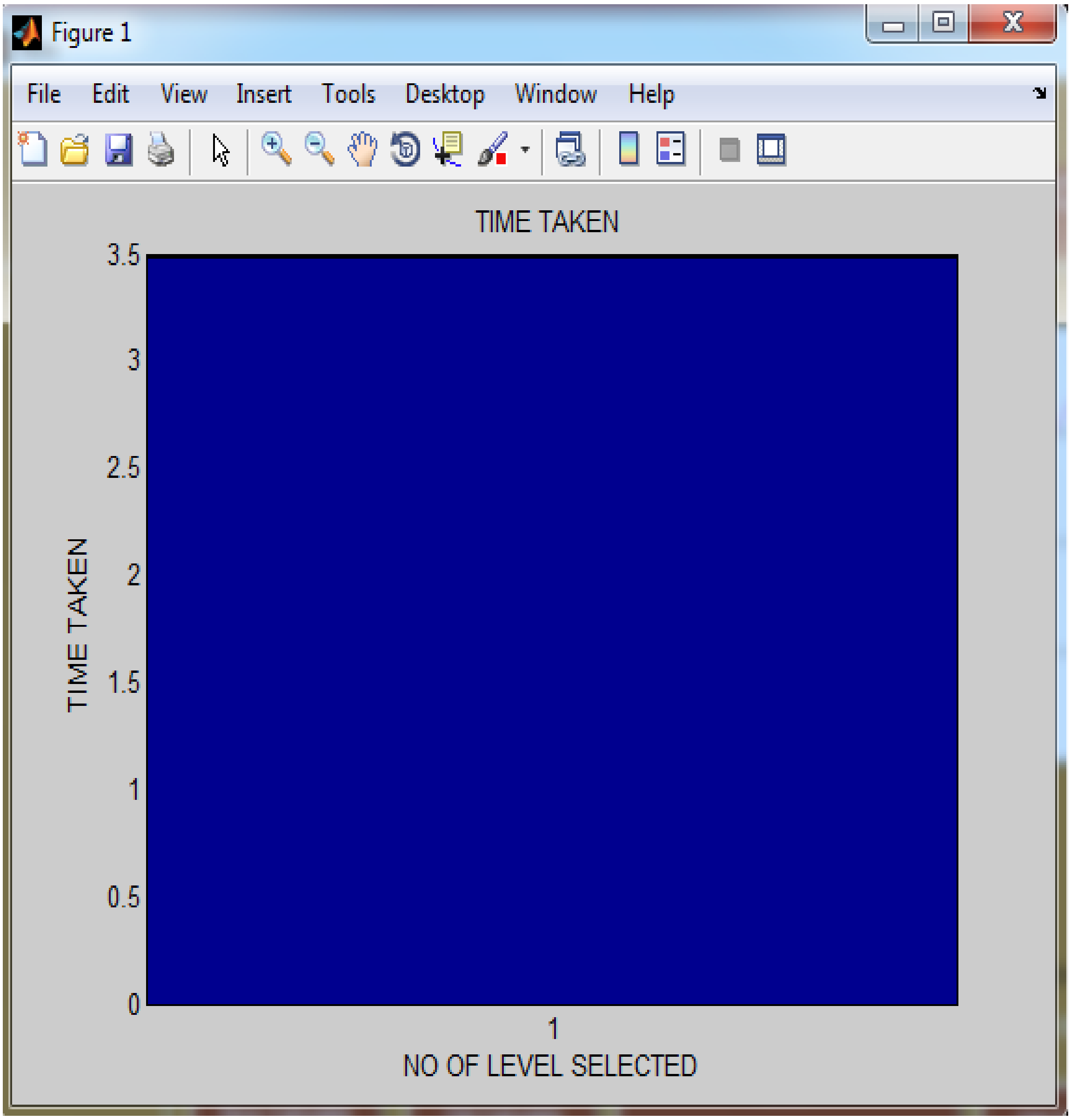
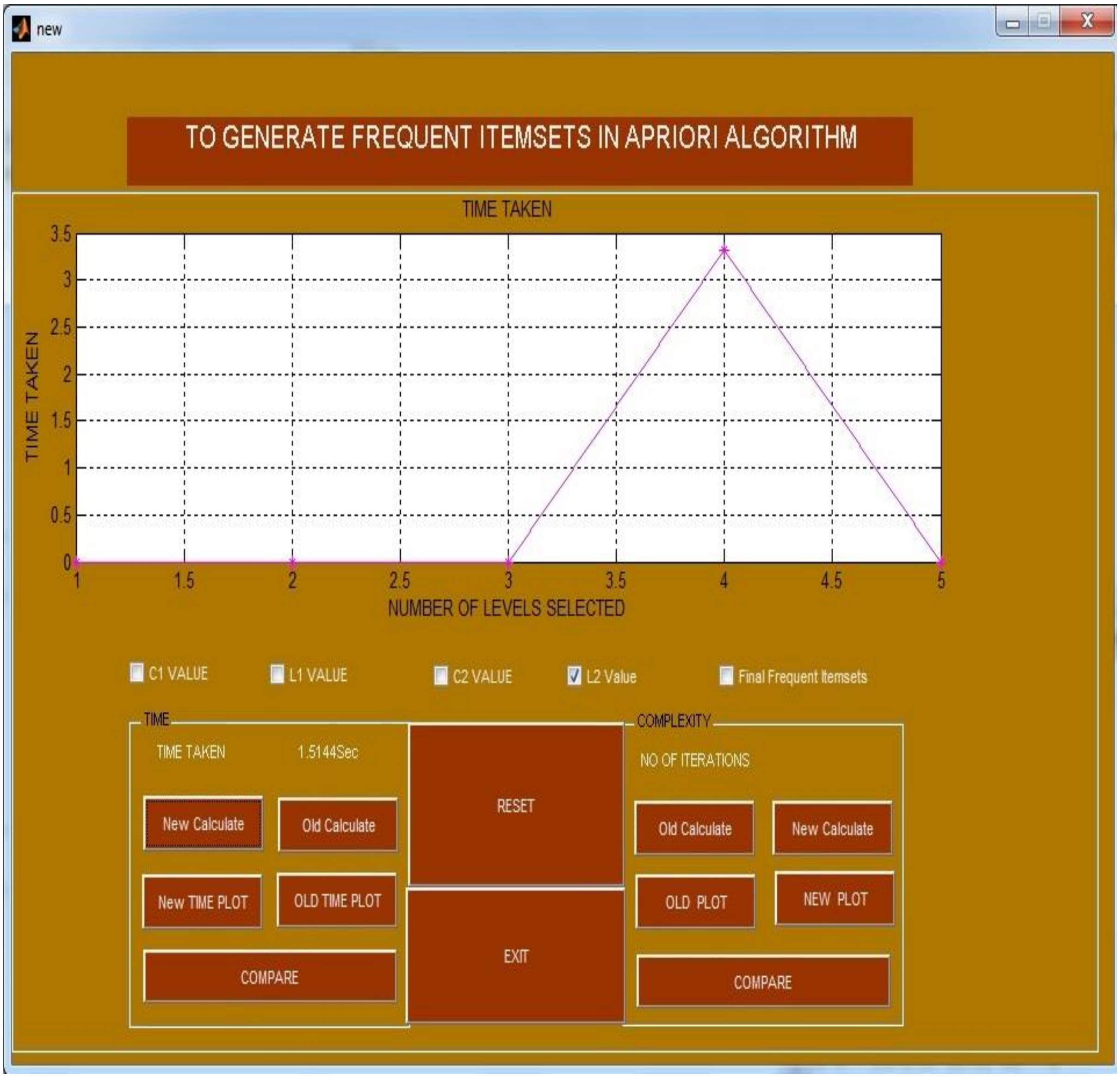


Figure 4.7: Plot of time analysis of existing algorithm for L2 value

As shown in figure 4.7, the performance analysis of Apriori’s algorithm is evaluated in terms of time taken to generate L2 value. The existing algorithm takes 3.48 seconds to generate L2 frequent itemset.



□ C1 VALUE □ L1 VALUE □ C2 VALUE ☑ L2 Value □ Final Frequent Itemsets

TIME TAKEN 1.5144Sec

Figure 4.8: Time analysis of enhanced algorithm for L2 value

As illustrated in figure 4.8, the performance analysis of Apriori’s algorithm is evaluated in terms of time taken to generate L2 value. The enhanced algorithm takes 1.51 seconds to generate L2 frequent itemset.

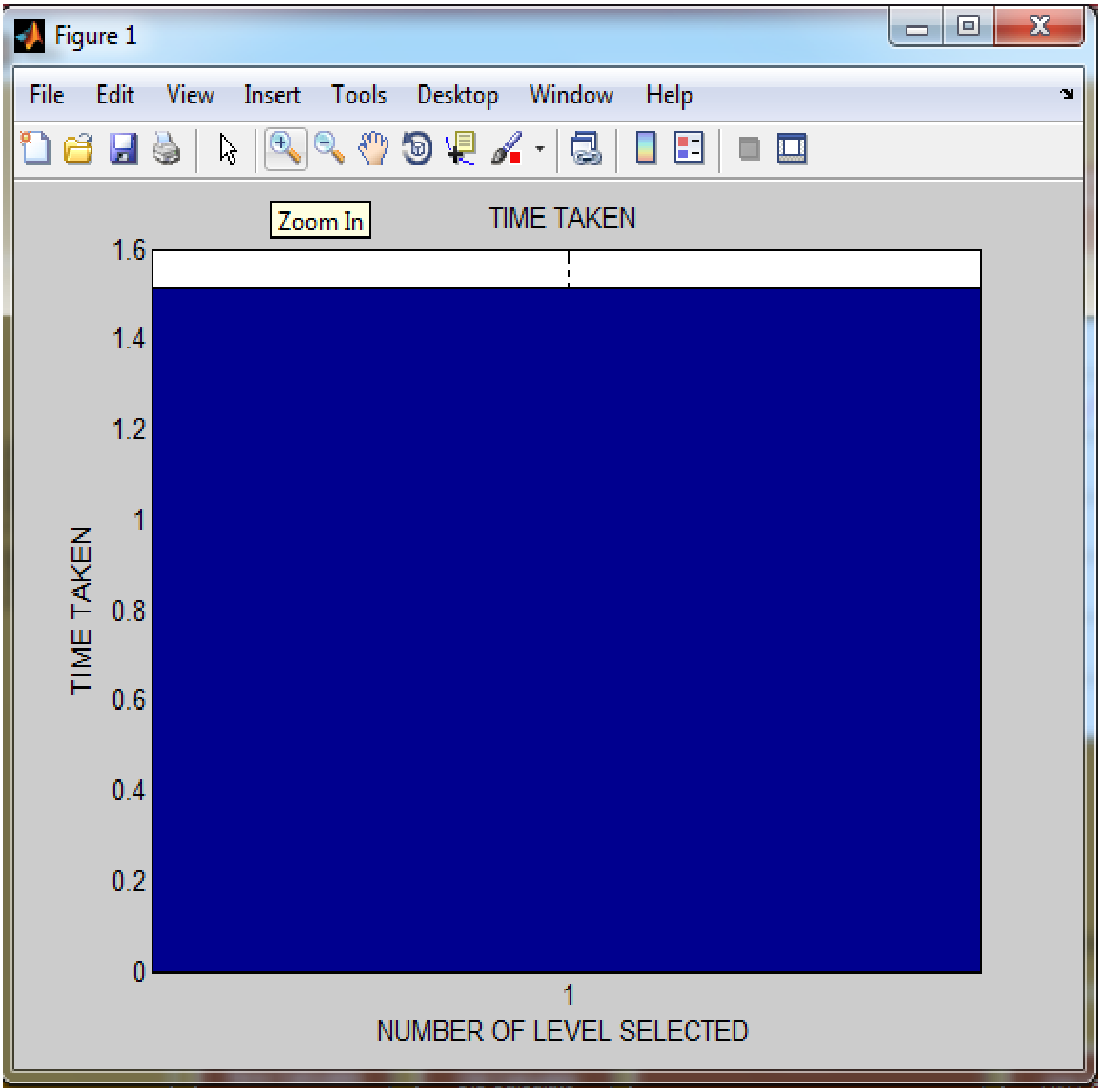


Figure 4.9: Plot of time analysis of enhanced algorithm for L2 value

As shown in figure 4.9, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate L2 value. The enhanced algorithm takes 1.51 seconds to generate L2 frequent itemset.

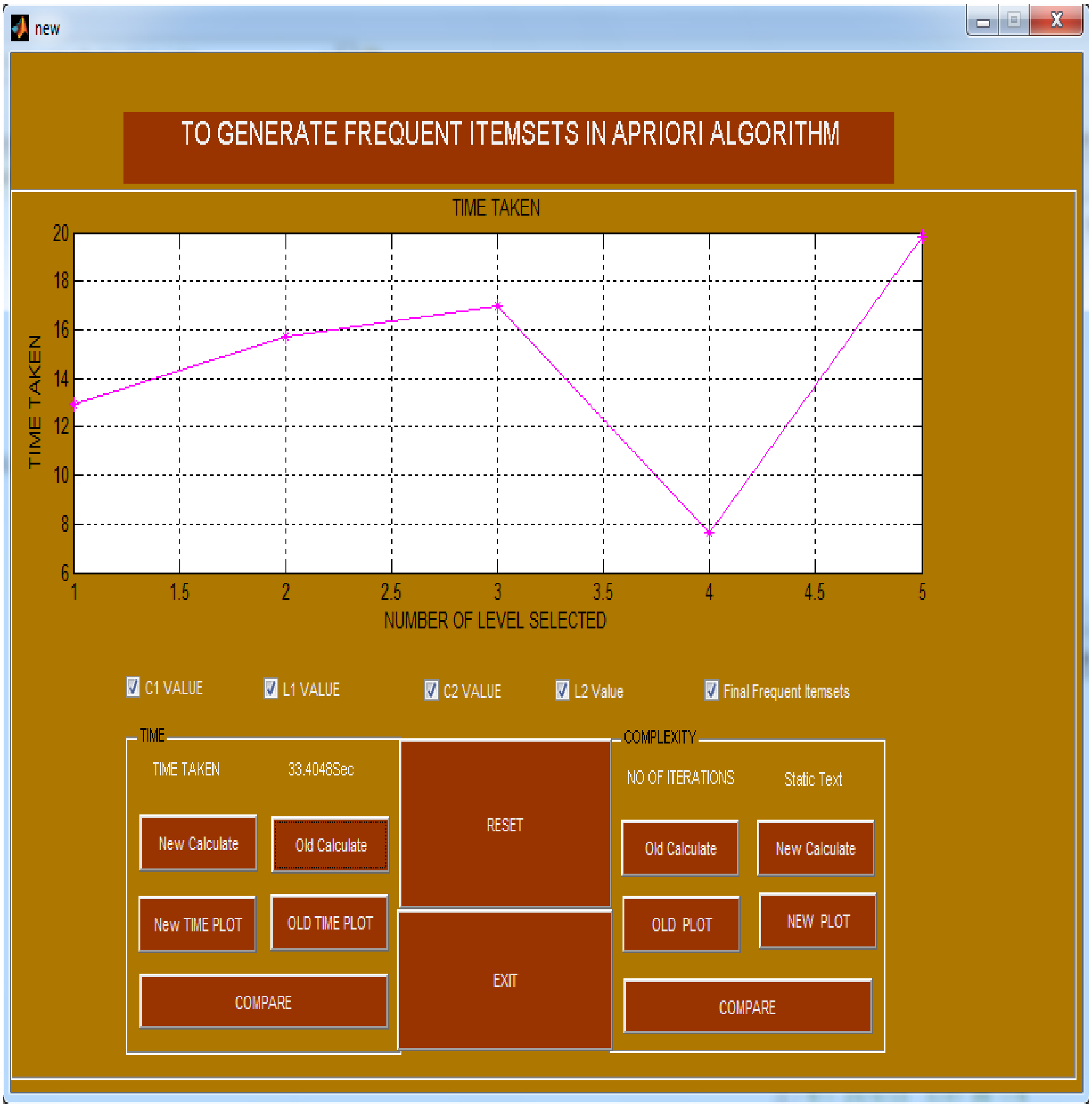


Figure 4.10: Time analysis of existing algorithm for final frequent itemsets

As illustrated in figure 4.10, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate final frequent itemsets. The existing algorithm takes 33.40 seconds to generate final frequent itemsets.

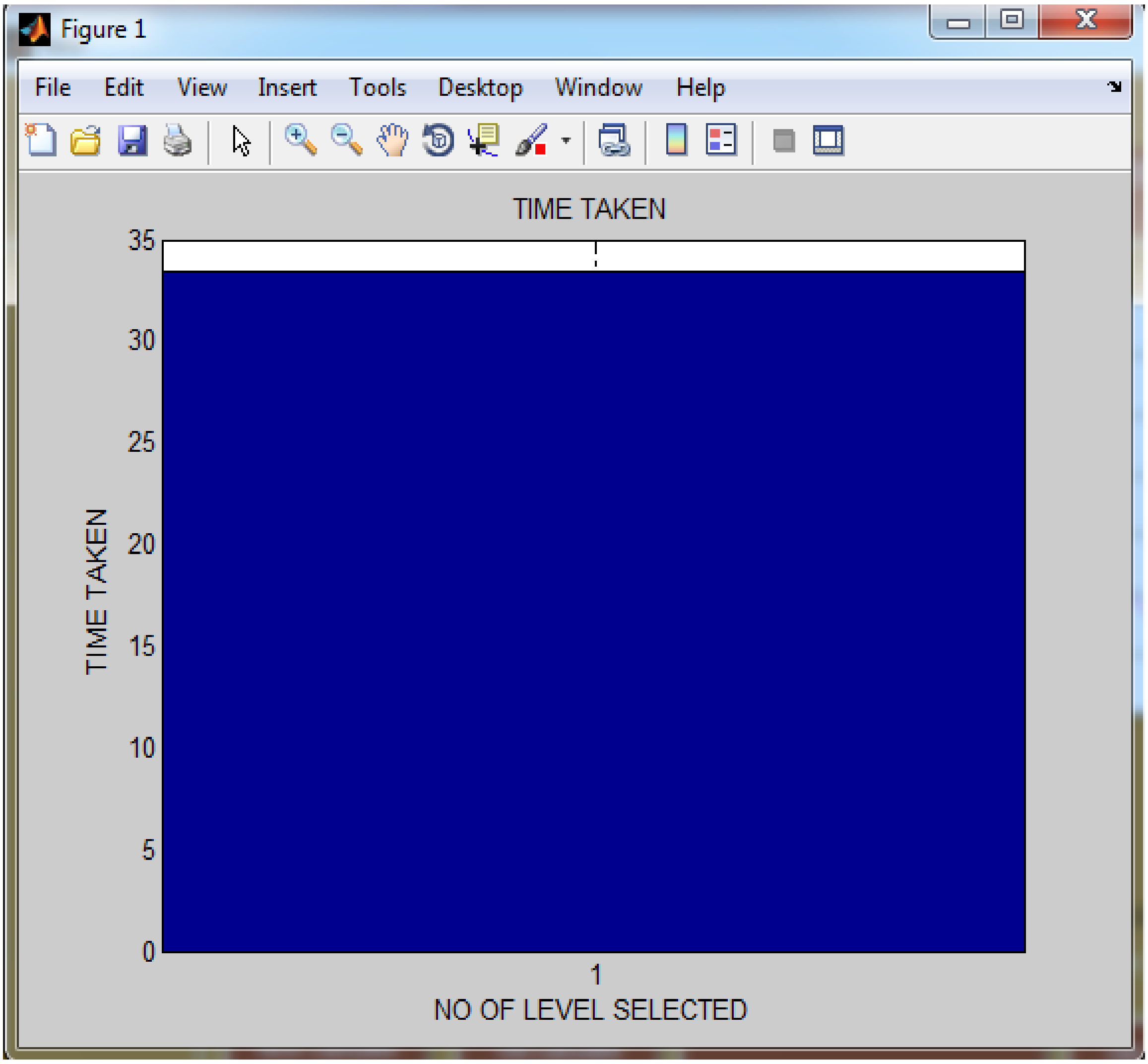
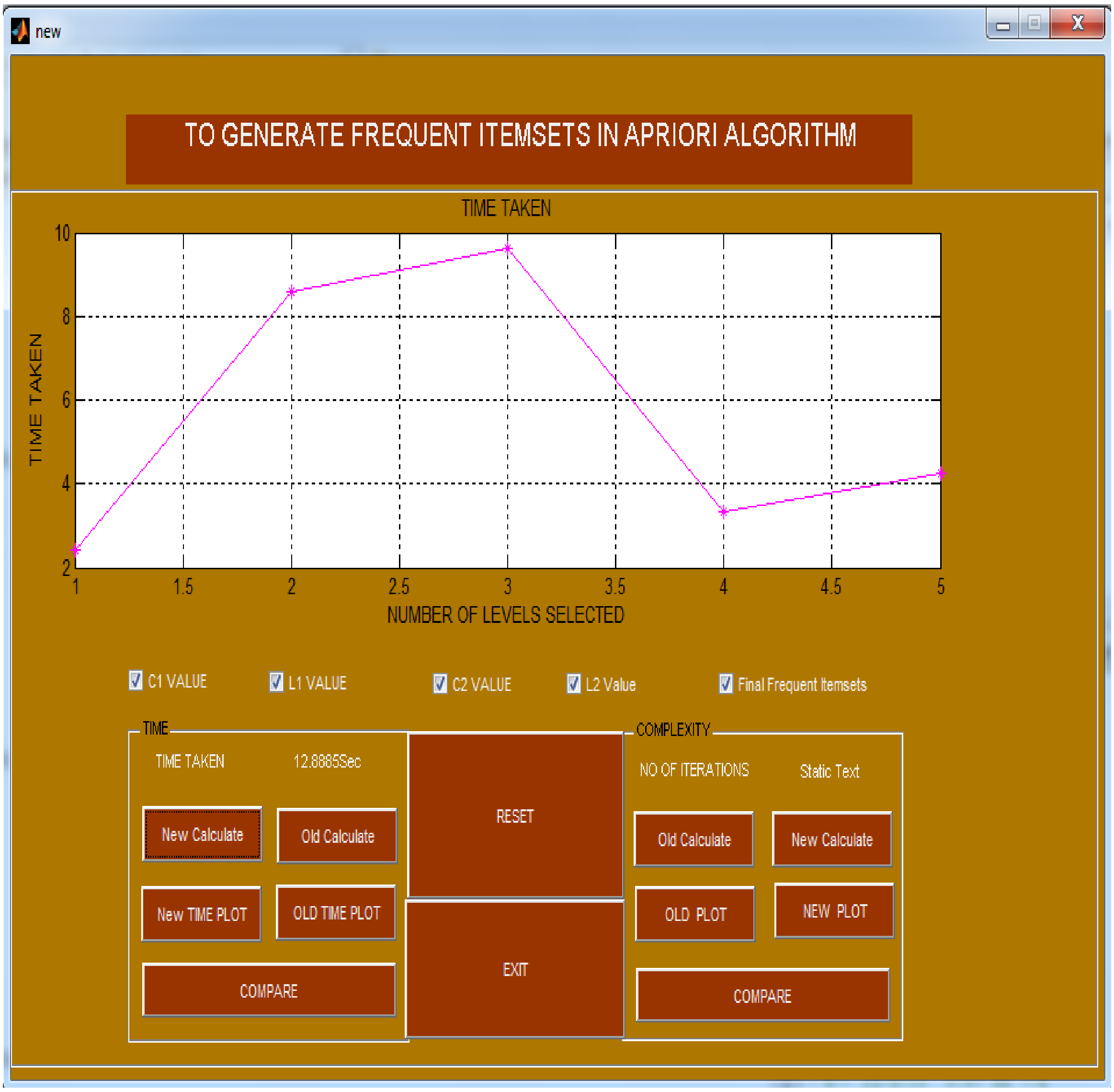


Figure 4.11: Plot of time analysis of existing algorithm for final frequent itemsets

As illustrated in figure 4.11, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate final frequent itemsets. The existing algorithm takes 33.40 seconds to generate final frequent itemsets.



☑ C1 VALUE ☑ L1 VALUE ☑ C2 VALUE ☑ L2 Value ☑ Final Frequent Itemsets

TIME TAKEN 12.8885Sec

NO OF ITERATIONS Static Text

Figure 4.12: Time analysis of enhanced algorithm for final frequent itemsets

As illustrated in figure 4.12, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate final frequent itemsets. The enhanced algorithm takes 12.88 seconds to generate final frequent itemsets.

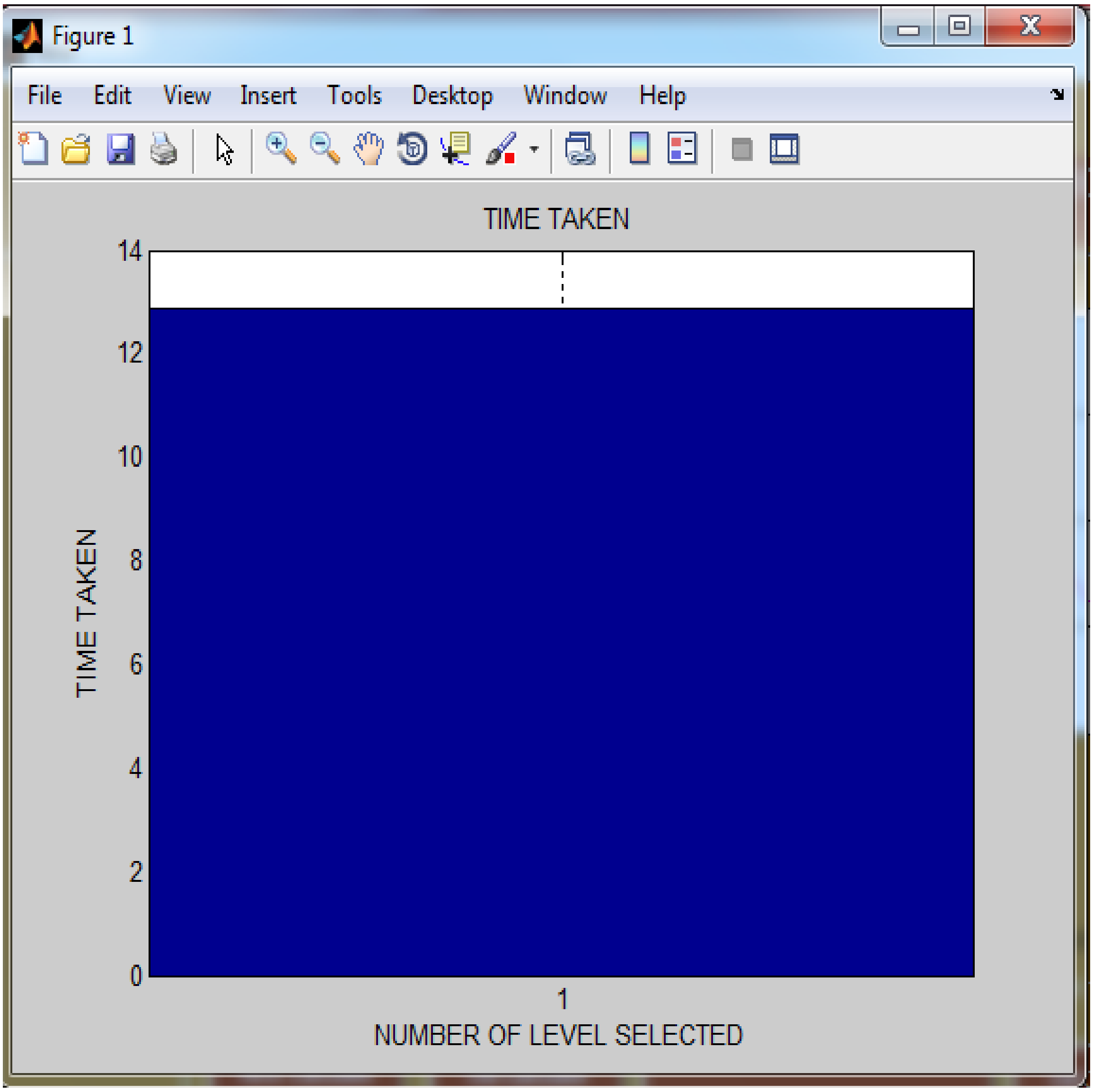
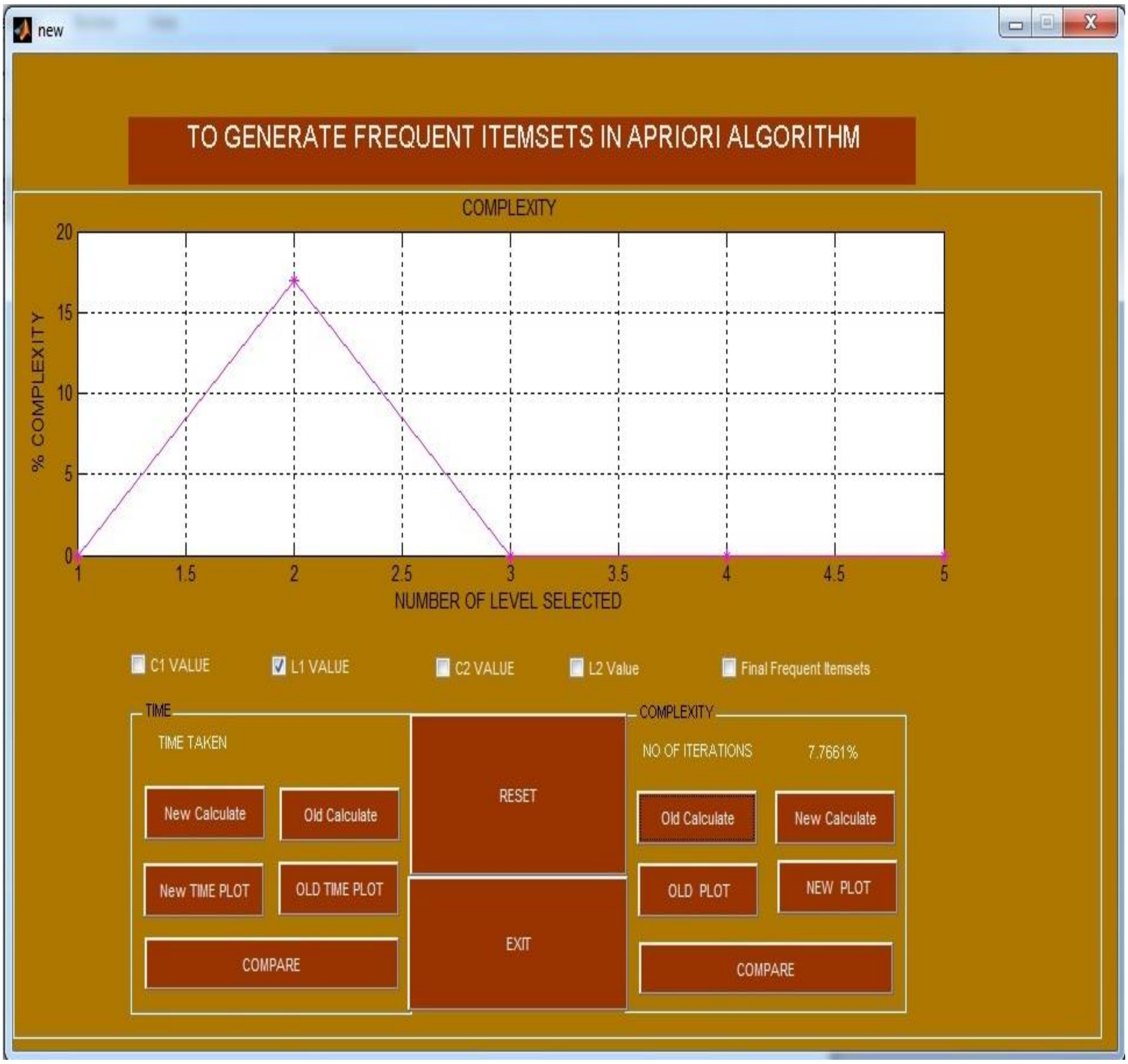


Figure 4.13: Plot of time analysis of enhanced algorithm for final frequent itemsets

As illustrated in figure 4.13, the performance analysis of Apriori's algorithm is evaluated in terms of time taken to generate final frequent itemsets. The enhanced algorithm takes 12.88 seconds to generate final frequent itemsets.



□ C1 VALUE ☑ L1 VALUE □ C2 VALUE □ L2 Value □ Final Frequent Itemsets

NO OF ITERATIONS 7.7661%

Figure 4.14: Complexity analysis of existing algorithm for L1 value

As illustrated in figure 4.14, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate L1 value. The existing algorithm takes 7.76 percent to generate L1 frequent itemset.

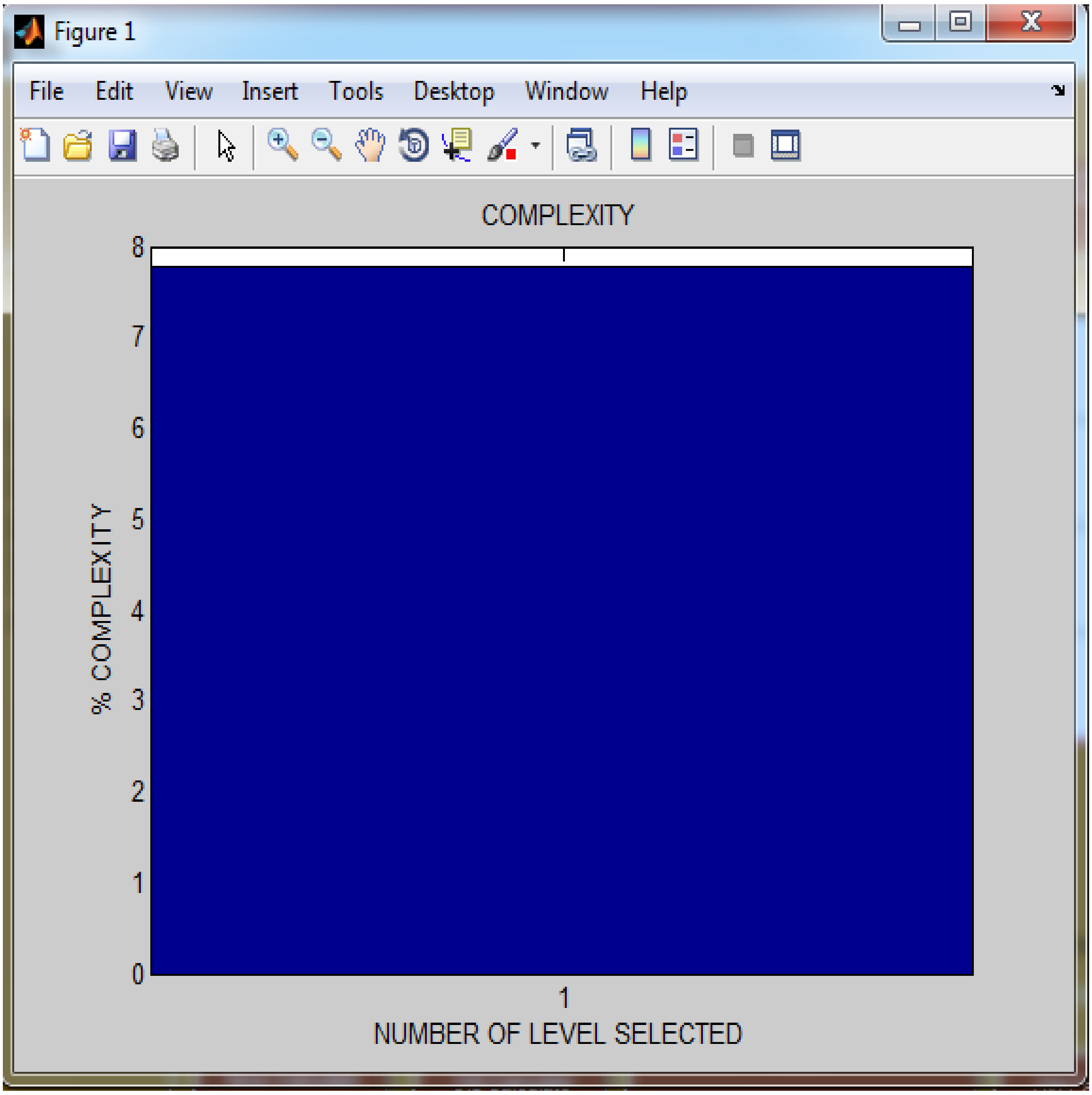
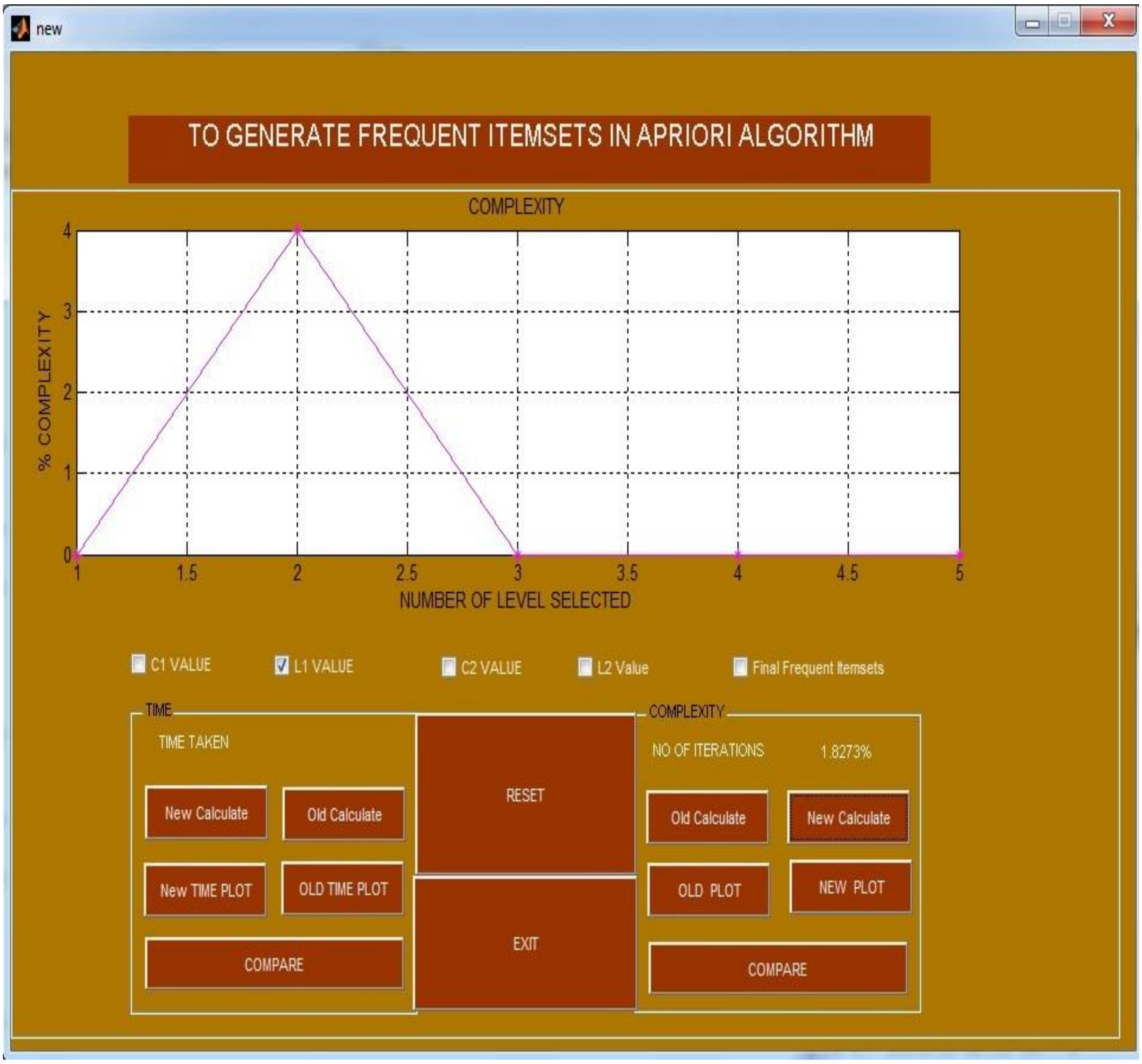


Figure 4.15: Plot of complexity analysis of existing algorithm for L1 value

As shown in figure 4.15, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate L1 value. The existing algorithm takes 7.76 percent to generate L1 frequent itemset.



NO OF ITERATIONS 1.8273%

Figure 4.16: Complexity analysis of enhanced algorithm for L1 value

As illustrated in figure 4.16, the performance analysis of Apriori’s algorithm is evaluated in terms of complexity to generate L1 value. The enhanced algorithm takes 1.82 percent to generate L1 frequent itemset.

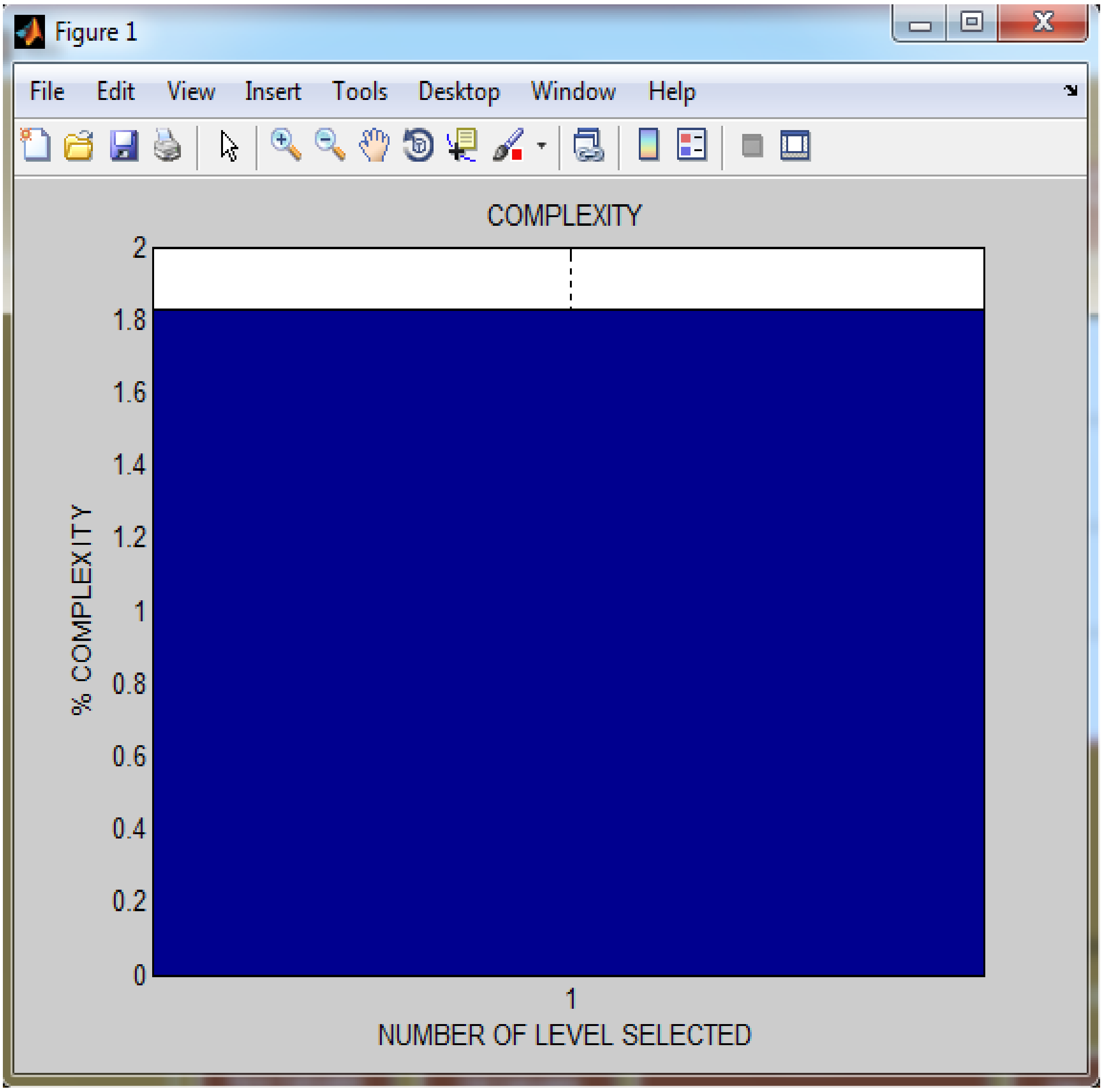
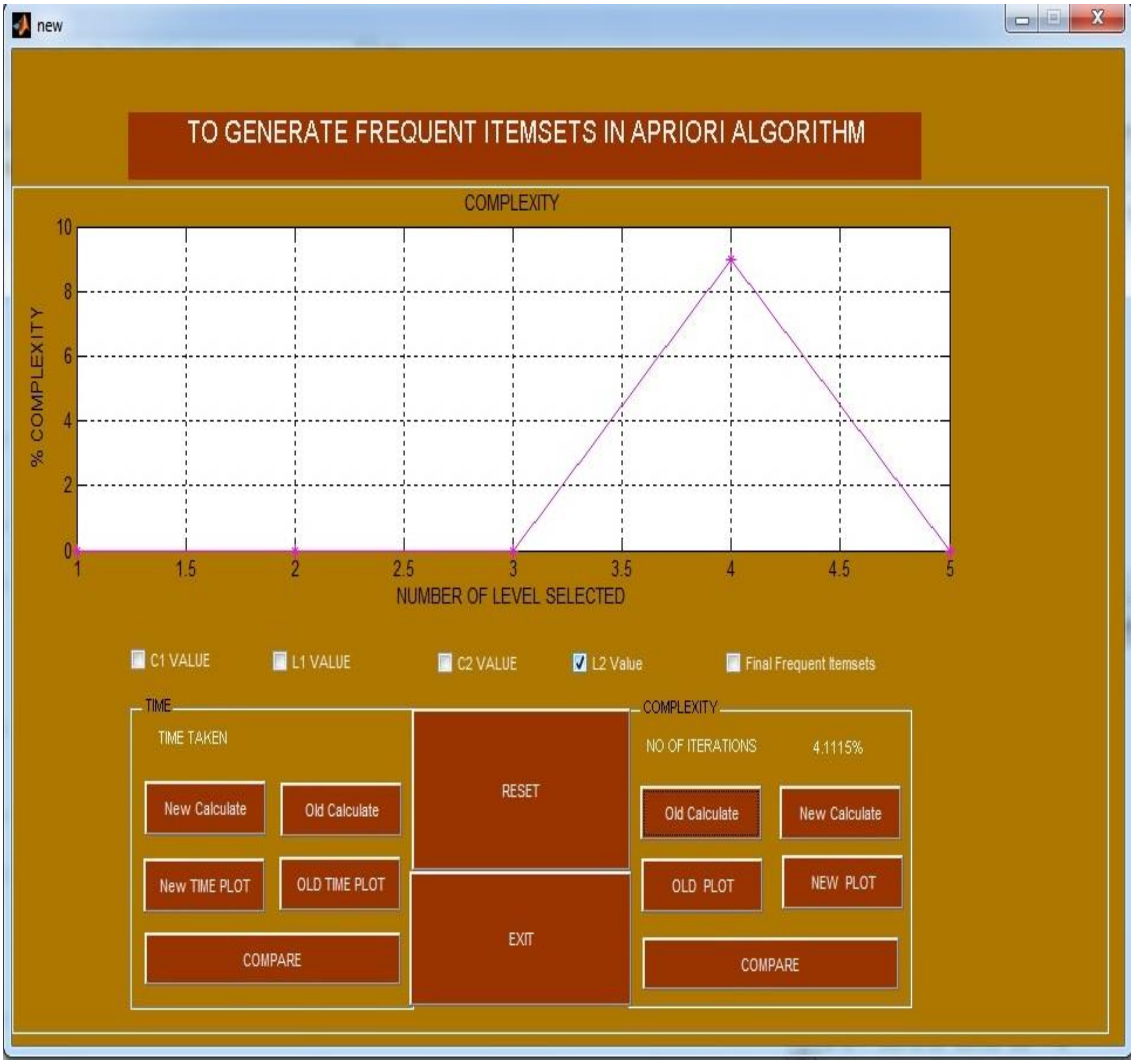


Figure 4.17: Plot of complexity analysis of enhanced algorithm for L1 value

As shown in figure 4.17, the performance analysis of Apriori’s algorithm is evaluated in terms of complexity to generate L1 value. The enhanced algorithm takes 1.82 percent to generate L1 frequent itemset.



□ C1 VALUE □ L1 VALUE □ C2 VALUE ☑ L2 Value □ Final Frequent Itemsets

NO OF ITERATIONS 4.1115%

Figure 4.18: Complexity analysis of existing algorithm for L2 value

As illustrated in figure 4.18, the performance analysis of Apriori’s algorithm is evaluated in terms of complexity to generate L2 value. The existing algorithm takes 4.11 percent to generate L2 frequent itemset.

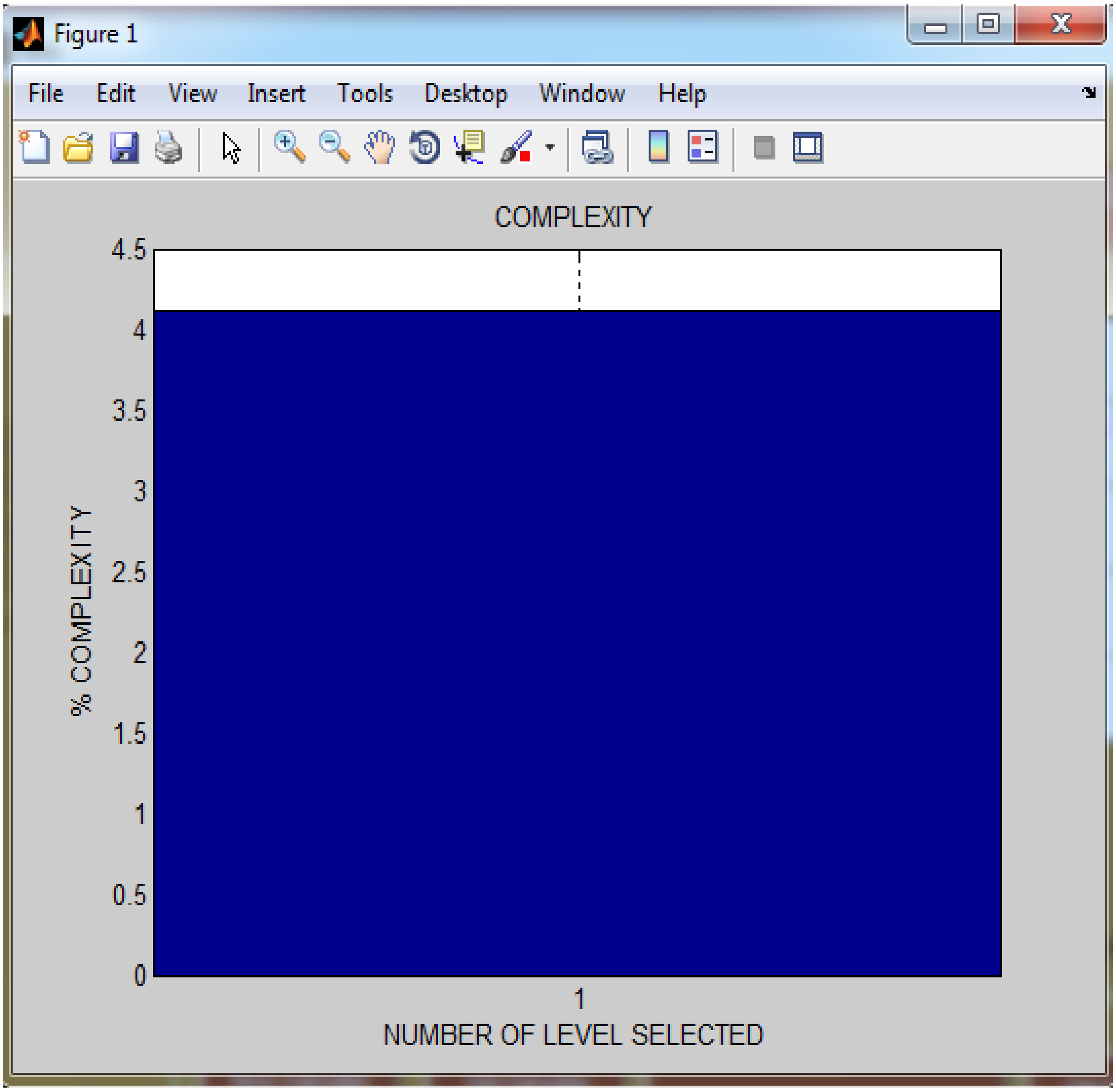
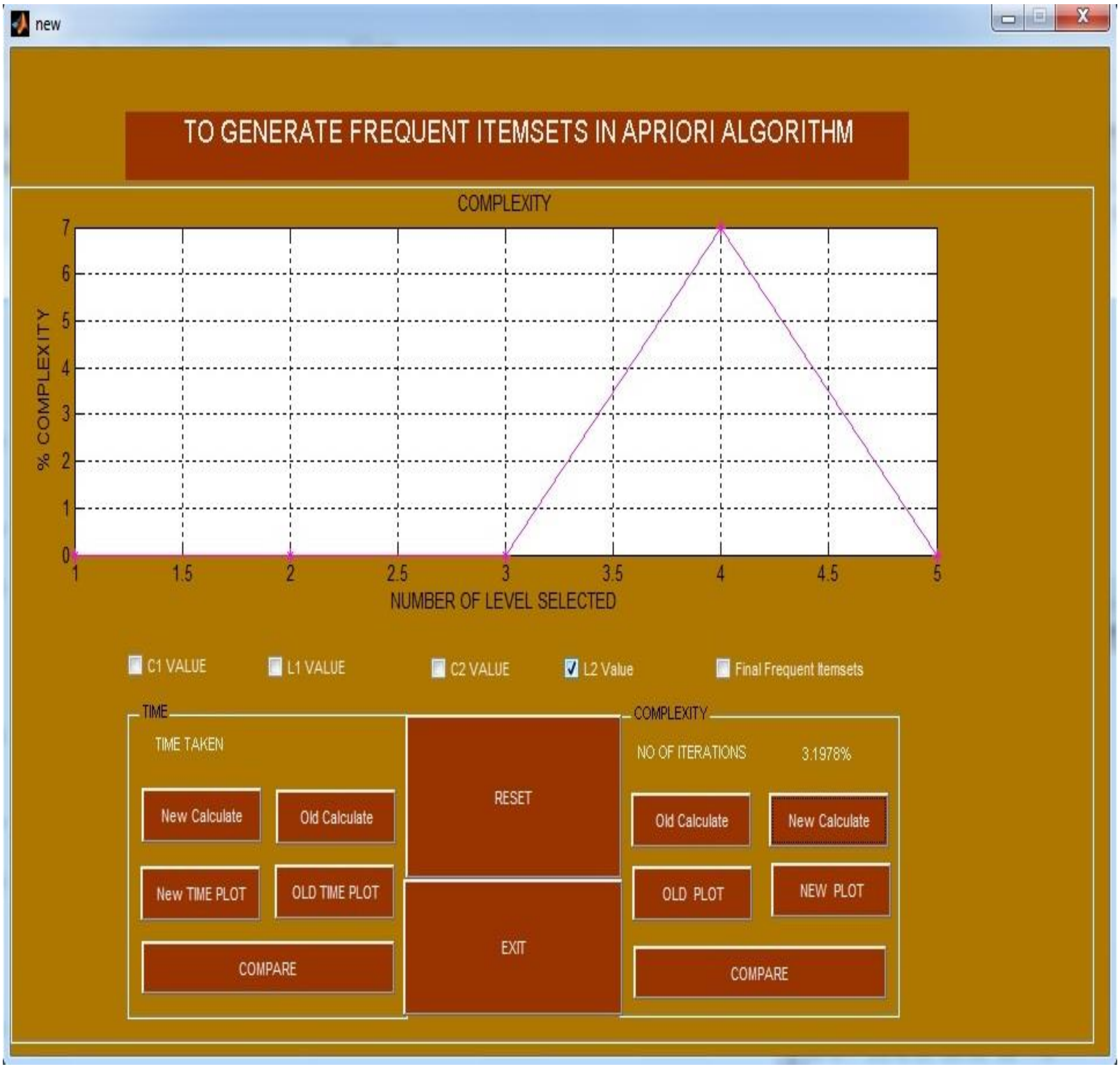


Figure 4.19: Plot of complexity analysis of existing algorithm for L2 value

As shown in figure 4.19, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate L2 value. The existing algorithm takes 4.11 percent to generate L2 frequent itemset.



□ C1 VALUE □ L1 VALUE □ C2 VALUE ☑ L2 Value □ Final Frequent Itemsets

NO OF ITERATIONS 3.1978%

Figure 4.20: Complexity analysis of enhanced algorithm for L2 value

As illustrated in figure 4.20, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate L2 value. The enhanced algorithm takes 3.19 percent to generate L2 frequent itemset.

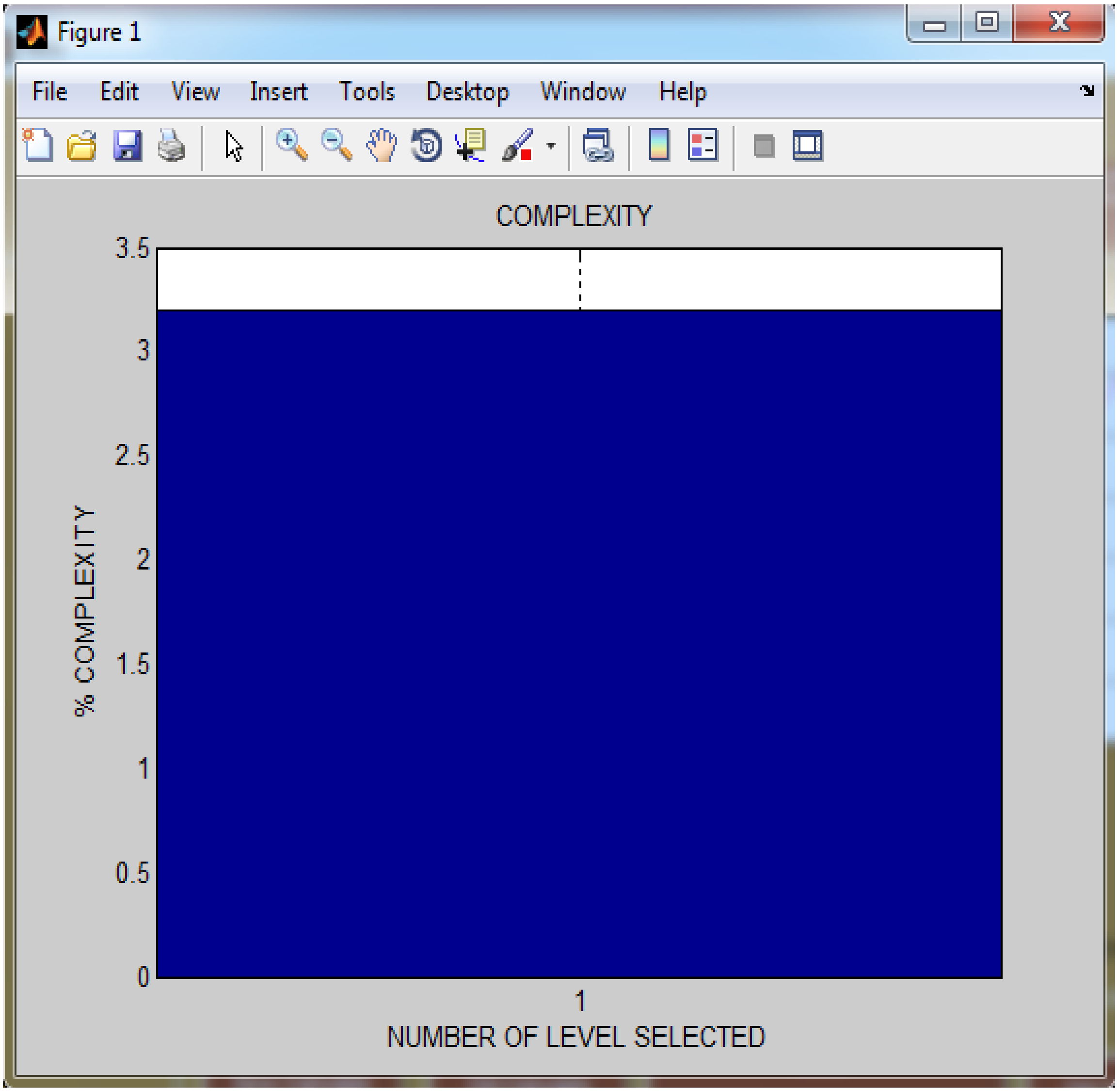


Figure 4.21: Plot of complexity analysis of enhanced algorithm for L2 value

As shown in figure 4.21, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate L2 value. The enhanced algorithm takes 3.19 percent to generate L2 frequent itemset.

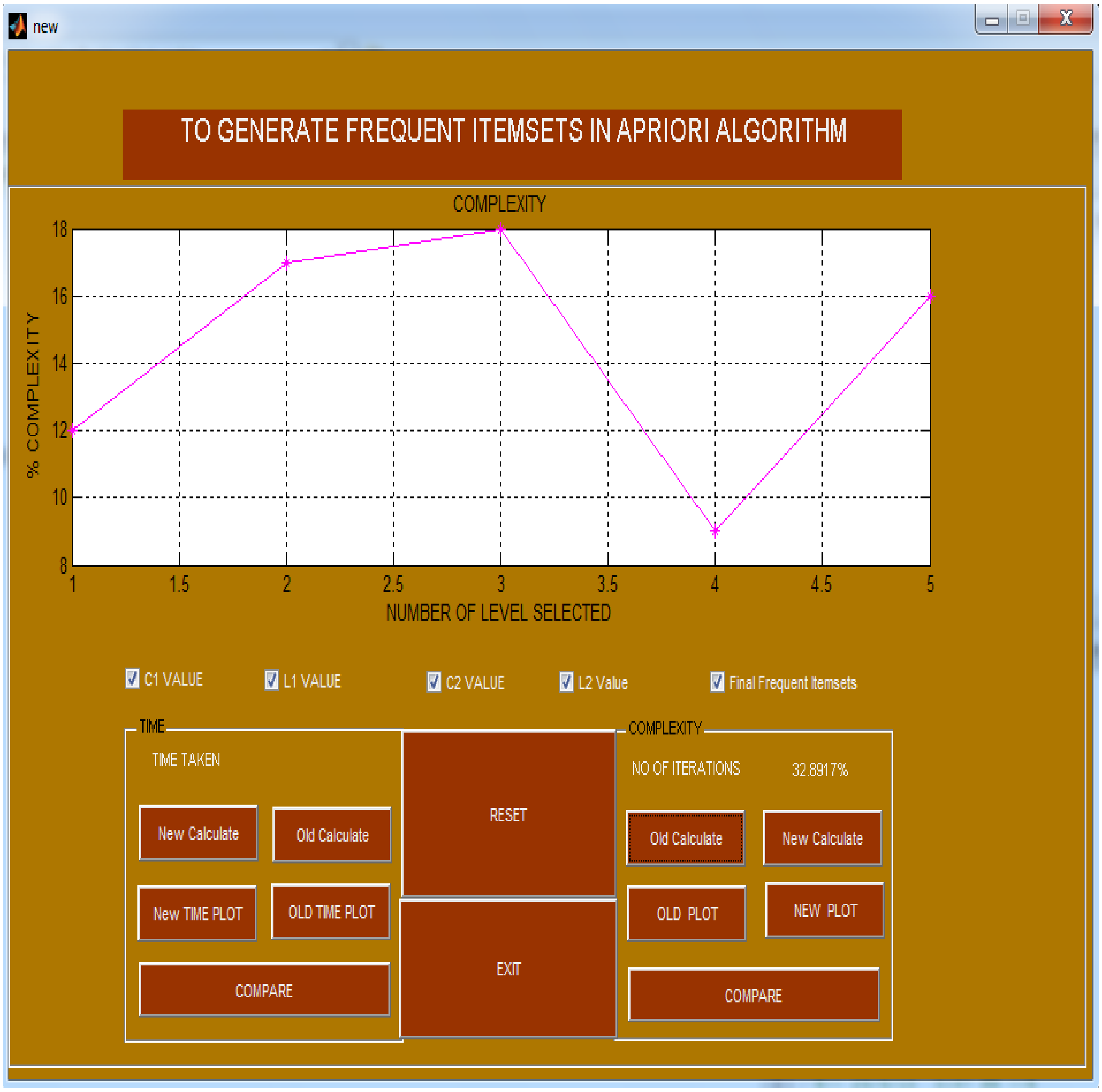


Figure 4.22: Complexity analysis of existing algorithm for final frequent itemsets

As illustrated in figure 4.22, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate final frequent itemsets. The existing algorithm takes 32.89 percent to generate final frequent itemsets.

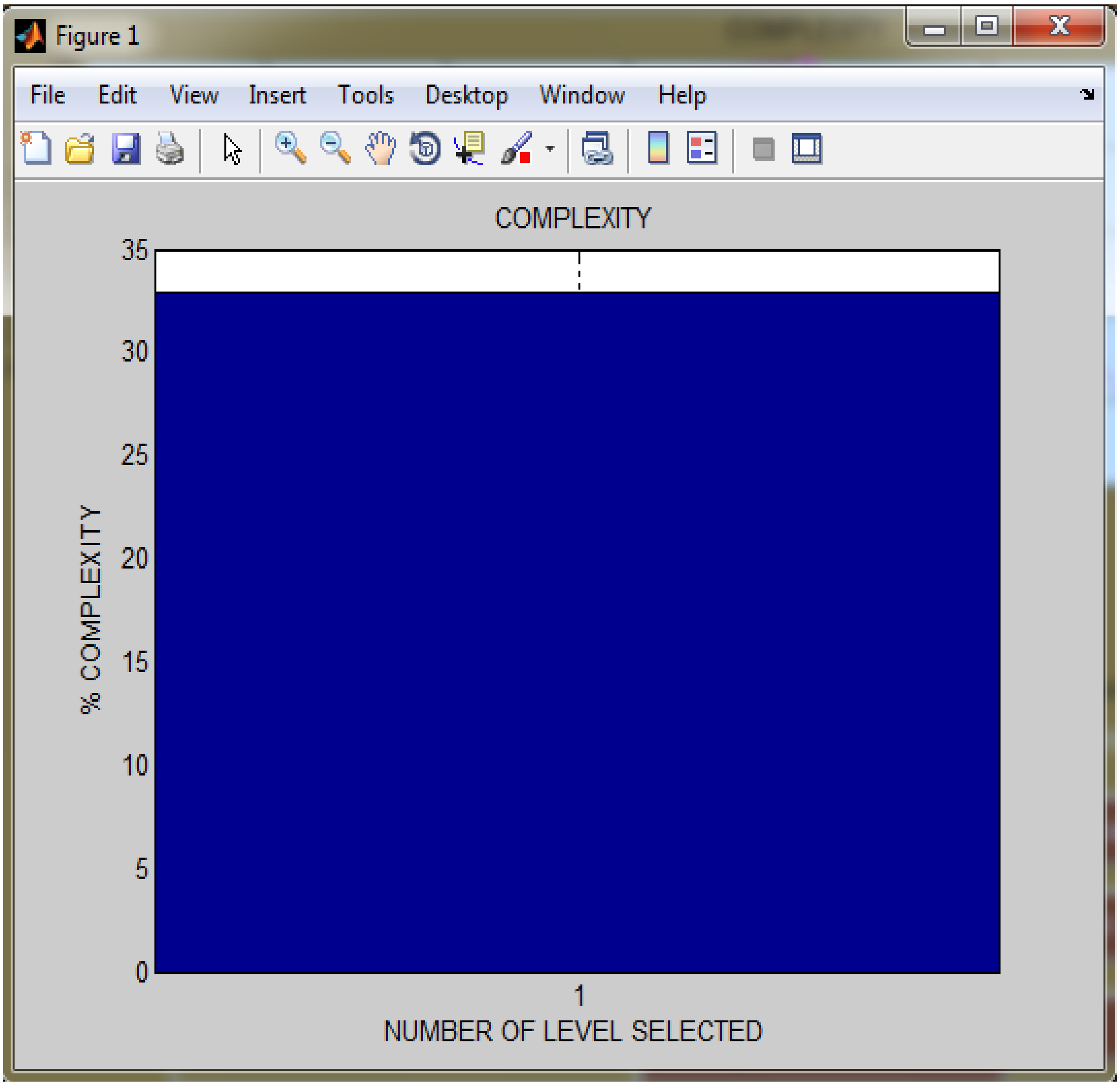
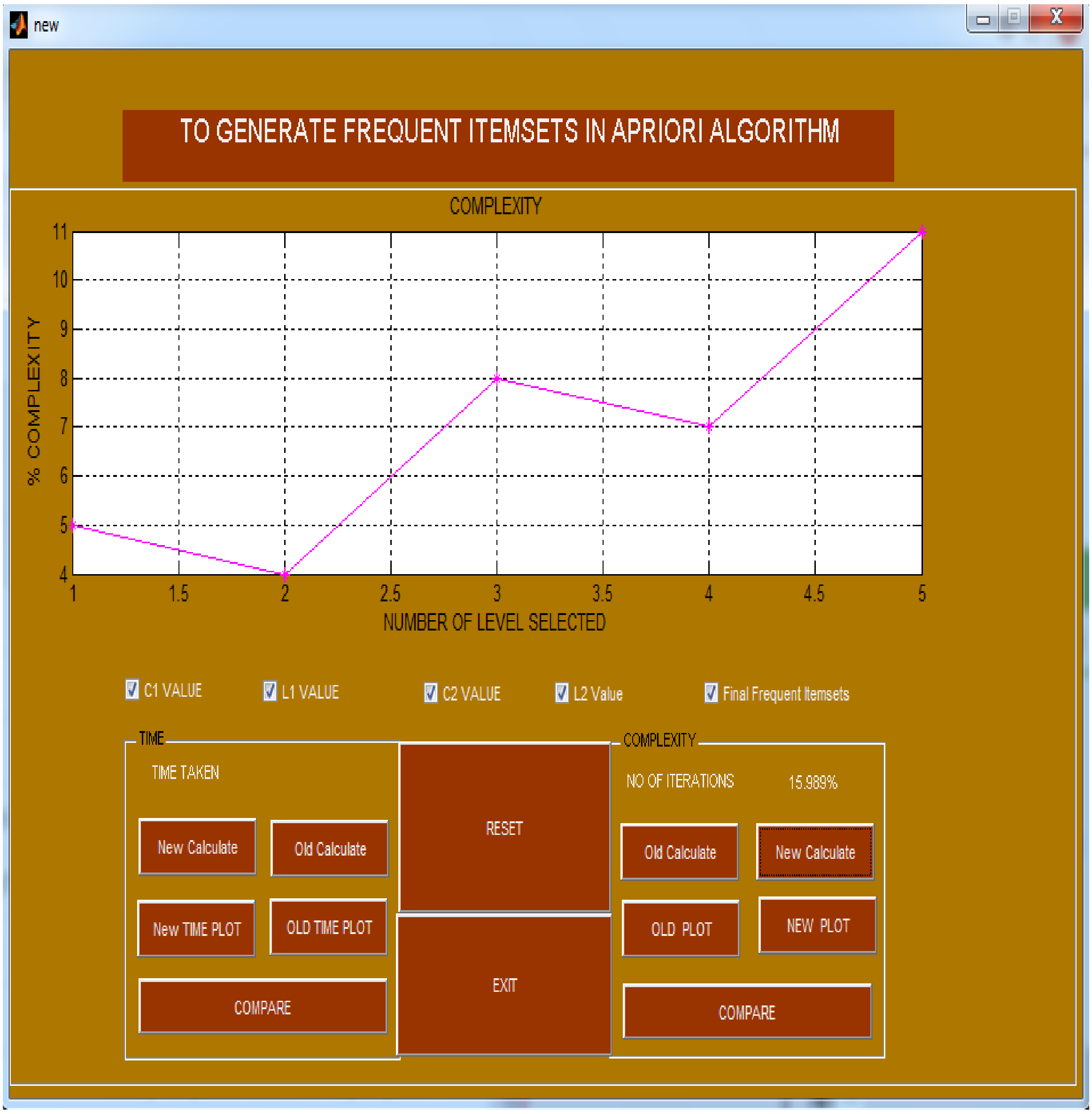


Figure 4.23: Plot of complexity analysis of existing algorithm for final frequent itemsets

As illustrated in figure 4.23, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate final frequent itemsets. The existing algorithm takes 32.89 percent to generate final frequent itemsets.



TO GENERATE FREQUENT ITEMSETS IN APRIORI ALGORITHM

☑ C1 VALUE ☑ L1 VALUE ☑ C2 VALUE ☑ L2 Value ☑ Final Frequent Itemsets

NO OF ITERATIONS 15.989%

Figure 4.24: Complexity analysis of enhanced algorithm for final frequent itemsets

As illustrated in figure 4.24, the performance analysis of Apriori’s algorithm is evaluated in terms of complexity to generate final frequent itemsets. The enhanced algorithm takes 15.98 percent to generate final frequent itemsets.

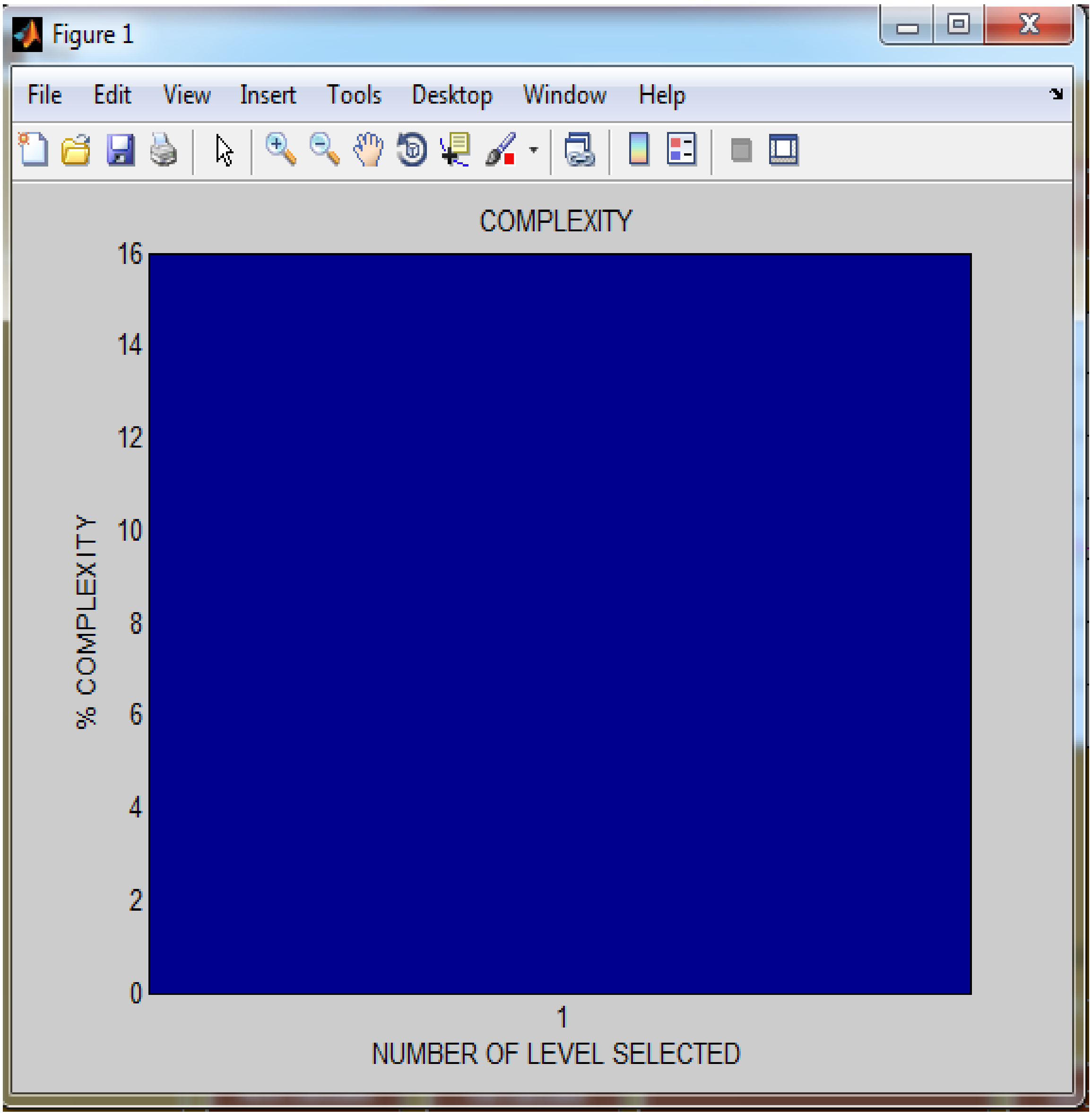


Figure 4.25: Plot of complexity analysis of enhanced algorithm for final frequent itemset

As illustrated in figure 4.25, the performance analysis of Apriori's algorithm is evaluated in terms of complexity to generate final frequent itemsets. The enhanced algorithm takes 15.98 percent to generate final frequent itemsets.

6.2 Performance Evaluation

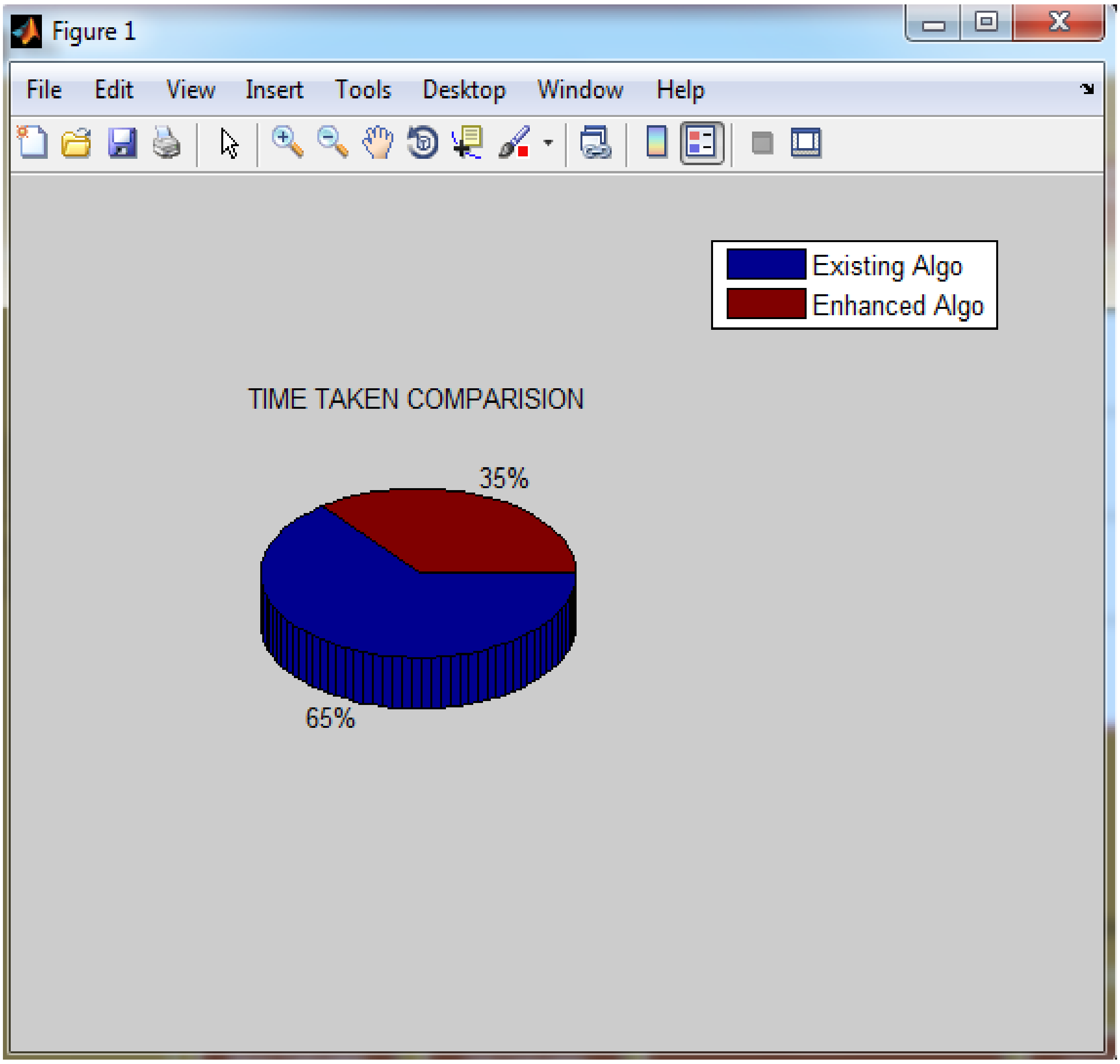


Figure 4.26: Comparison analysis of time for L1 value

In figure 4.26, the time analysis of existing and enhanced algorithm are shown. In the pie chart it clear that, 65 % time used by the existing algorithm and only 35 % time is taken by enhanced algorithm to generate L1 frequent itemset.

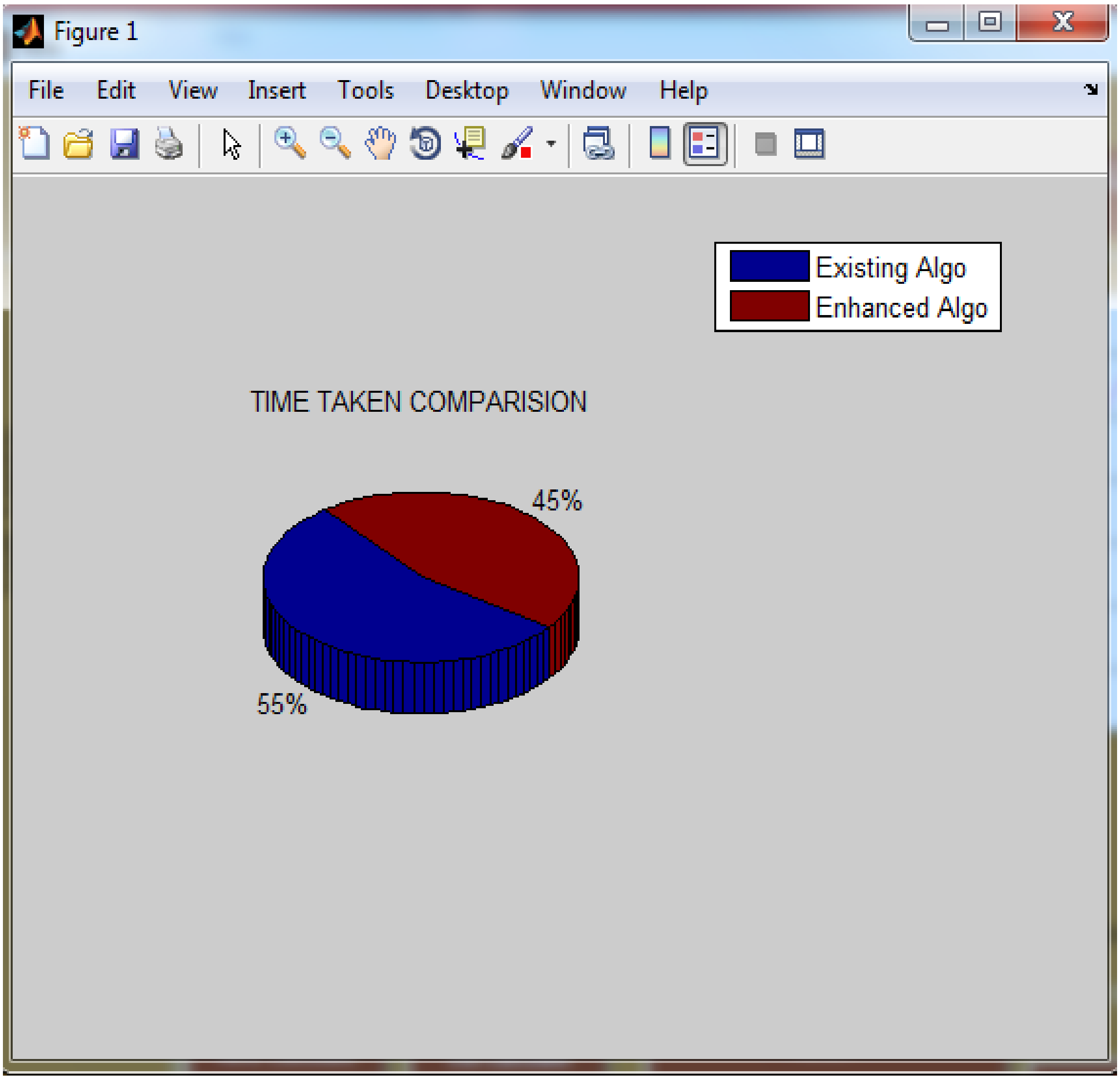


Figure 4.27: Comparison analysis of time for L2 value

In figure 4.27, the time analysis of existing and enhanced algorithm are shown. In the pie chart it is clear that, 55% time is used by the existing algorithm and only 45 % time is taken by enhanced algorithm to generate L2 frequent itemset.

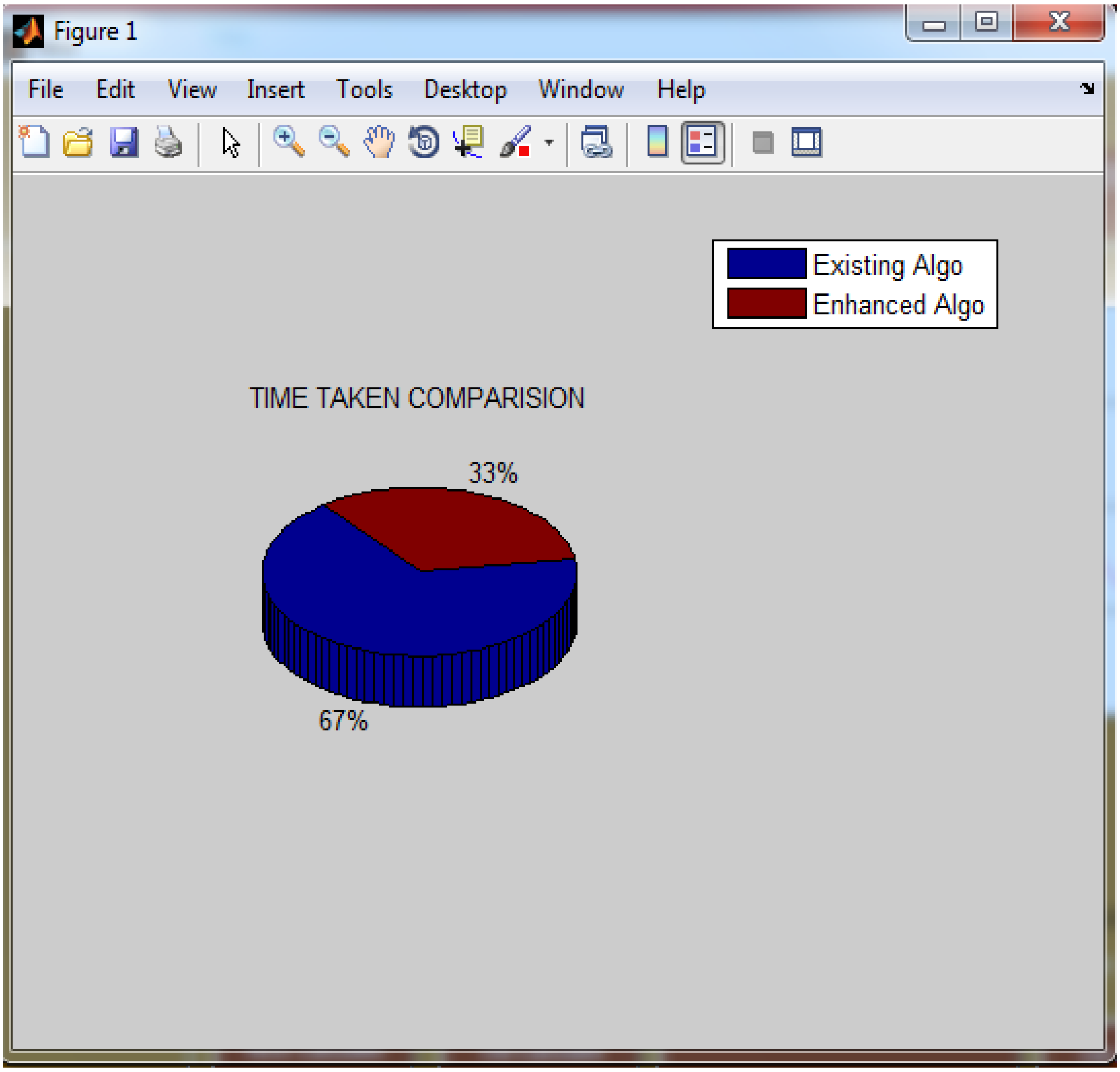


Figure 4.28: Comparison analysis of time for final frequent itemsets

In figure 4.28, the time analysis of existing and enhanced algorithm are shown. In the pie chart it clear that, 67 % time used by the existing algorithm and only 33 % time is taken by enhanced algorithm to generate final frequent itemsets.

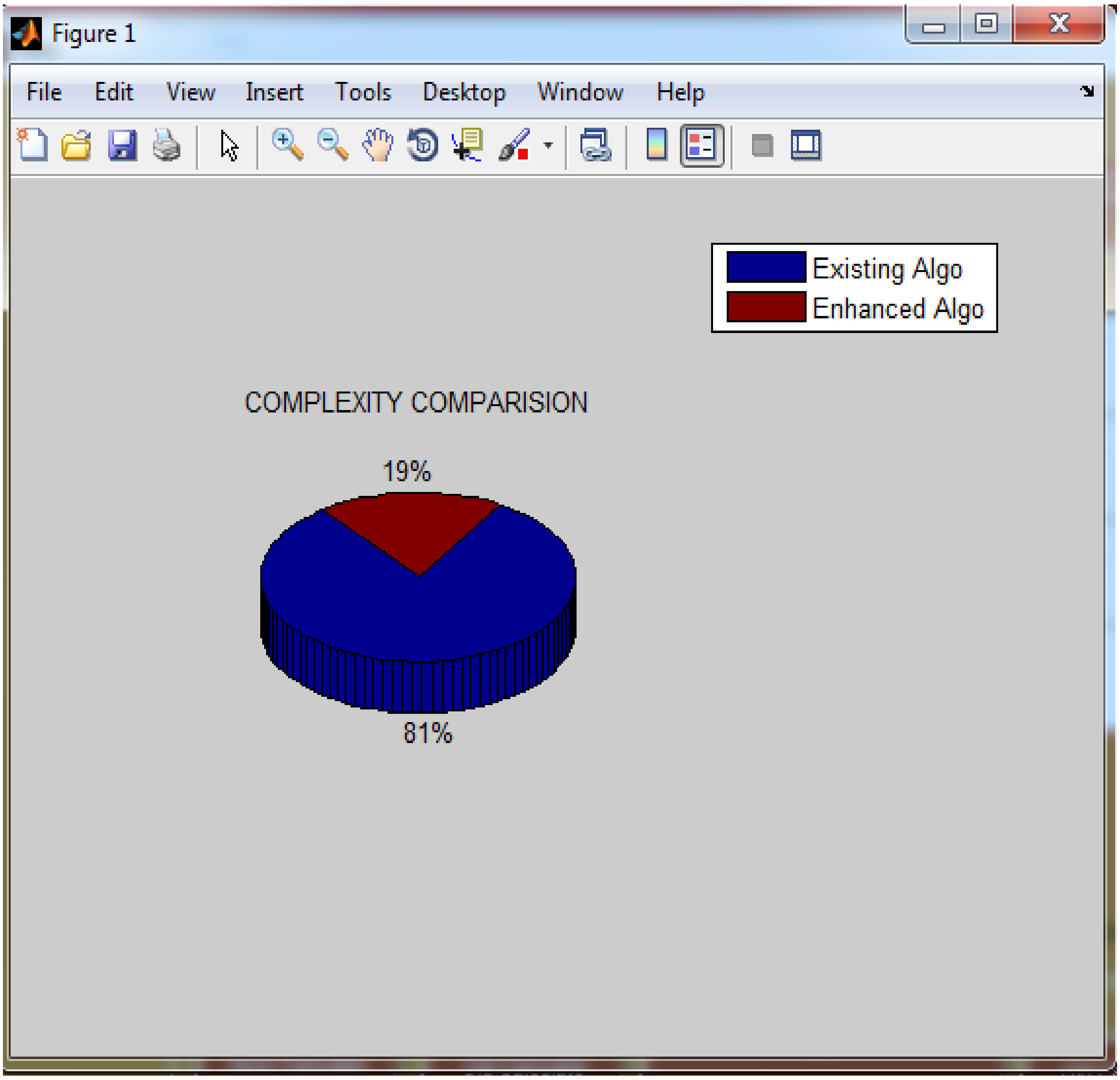


Figure 4.29: Comparison of complexity for L1 value

As illustrated in figure 4.29, the comparison is made between existing and enhanced algorithm in terms of complexity. The piechart shows that existing algorithm has 81 % complexity and new algorithm has 19 % complexity to generate L1 frequent itemset.

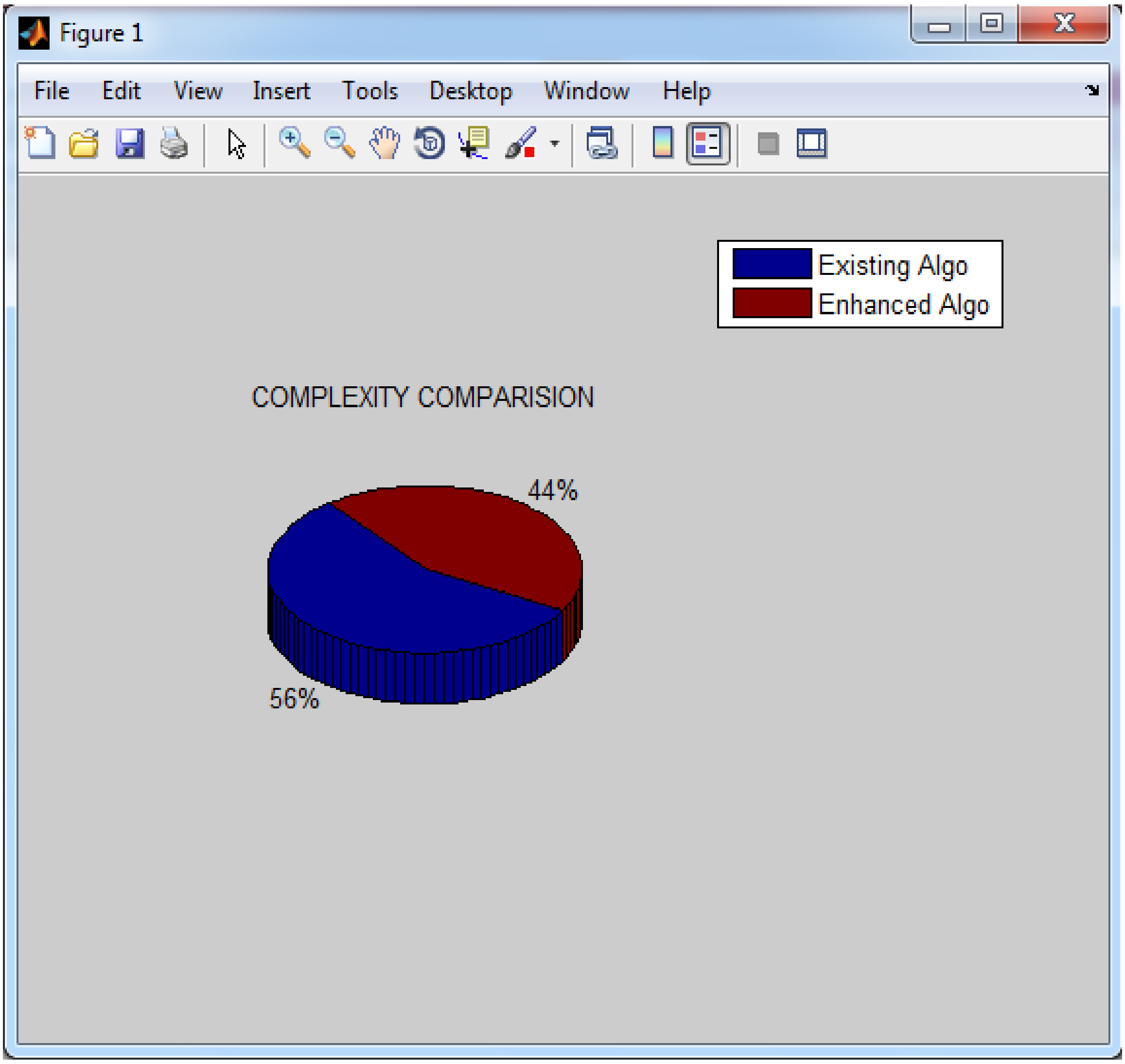


Figure 4.30: Comparison of complexity for L2 value

As illustrated in figure 4.30, the comparison is made between existing and enhanced algorithm in terms of complexity. The piechart shows that existing algorithm has 56% complexity and new algorithm has 44% complexity to generate L2 frequent itemset.

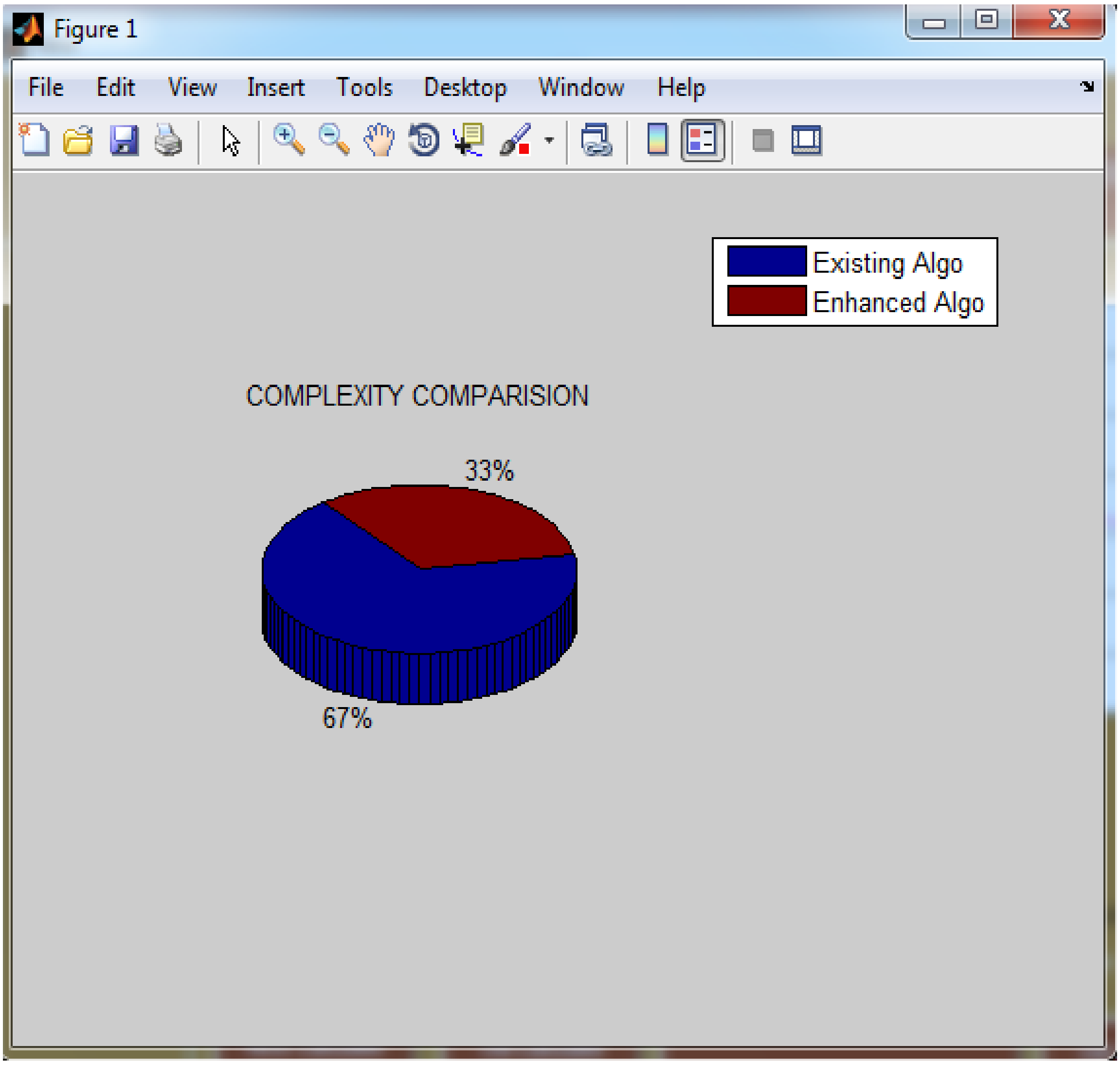


Figure 4.31: Comparison of complexity for final frequent itemsets

As illustrated in figure 4.31, the comparison is made between existing and enhanced algorithm in terms of complexity. The piechart shows that existing algorithm has 67% complexity and new algorithm has 33% complexity to generate final frequent itemset.

Chapter 5

CONCLUSION AND FUTURE SCOPE

We have proposed enhancement to implement Apriori algorithm in MATALAB using Laplace equation to improve the efficiency of existing Apriori algorithm. The proposed algorithm reduces the execution time and complexity and is more reliable than existing Apriori algorithm. Laplacian matrix is the product of the incidence matrix with its transpose, so the Laplacian can be consider as a correlation matrix for the objects, but with correlation based only on connection. By comparing the results of existing and proposed algorithm, it is shown that proposed Apriori algorithm is better than existing Apriori algorithm as the time taken by proposed Apriori algorithm is less than existing Apriori algorithm and the complexity of the proposed Apriori algorithm is also less than existing Apriori algorithm. Further, the work may be extended by applying Laplace equation on Eclat algorithm to generate frequent itemsets.

Chapter 6

LIST OF REFERENCES

JOURNALS

Chanchal Yadav, S. W. (2013). An Approach to Improve Apriori Algorithm Based On Association rule Mining. *4th ICCCNT.* Tiruchengode, India: IEEE.

Gang FANG, J. X.-S.-P. (2010). *An Algorithm of Mining Spatial Topology Association Rules Based on Apriori.* China: IEEE.

Garg, M. S. (2013). Mining Efficient Association Rules Through Apriori Algorithm Using Attributes and Comparative Analysis of Various Association Rule Algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*.

Huan Wu, Z. L. (2009). An Improved Apriori-based Algorithm for Association Rules Mining. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. China: IEEE.

Jaishree Singh, H. R. (2013). Improving efficiency of apriori algorithm using Transaction reduction. *International Journal of Scientific and Research Publications*.

Kasamsan, S. A. (2010). *The Comparative of Boolean Algebra Compress and Apriori Rule Techniques for New Theoretic Association Rule Mining Model.* Bangkok, Thailand.

P.Uma, D. a. (2012). An Improved Frequent Pattern Algorithm for Mining Association Rules. *International Journal of Information and Communication Technology Research*.

Paul, R. a. (2010). Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery. *International Conference on Computing, Communication and Networking Technologies.* Coimbatore, India: IEEE.

R.Santhi, K. a. (2011). Using Hash Based Apriori Algorithm To Reduce The Candidate 2-Itemsets for Mining Association Rule. *Journal of Global Research in Computer Science*.

Rina Raval, I. j. (2013). Survey on several improved Apriori algorithms. *IOSR Journal of Computer Engineering*.

S.Bagga, N. a. (2014). Implementation of Apriori Algorithm in MATLAB using Attribute Affinity Matrix. *International Journal of Advanced Research in Computer Science and Software Engineering*.

S.Vijayarani, D. a. (2010). Privacy Preserving Data Mining Based on Association Rule- A Survey. *Proceedings of the International Conference on Communication and Computational Intelligence.* Tamilnadu, India.

Wanjun Yu, X. W. (2008). The Research of Improved Apriori Algorithm for Mining Association Rules. *11th IEEE International Conference on Communication Technology proceedings.* China: IEEE.

Wei-Min Ma, Z.-P. L. (2008). Two revised algorithms based on Apriori for mining association rules. *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics.* China: IEEE.

Xue-Gang Hu, D.-X. W.-P. (2004). The analysis on model of association rule mining based on concept lattice and Apriori algorithm. *Proceedings of the Third International conference on Machine learning and Cybernetics.* Shanghai: IEEE.

Yanfei Zhou, W. W. (2010). Mining Association Rules Based on an Improved Apriori Algorithm. *International Conference on Audio, Language and Image Processing.* China: IEEE.

Yang, S. (2012). Research and Application of Improved Apriori Algorithm to Electronic Commerce. *11th International Symposium on Distribued Computing and Applications to Bussiness, Engineering and Science.* China: IEEE.

BOOKS

Kamber, J. H. (2006). *Data Mining Concepts and Techniques.* ELSEVIER.

WEBSITES

www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/itemset_**apriori**.html

http://www.zentut.com/data-mining/data-mining-applications/

Chapter 7

APPENDIX

7.1 ABBREVIATIONS

KDD- Knowledge Discovery In Databases

DHP- Direct Hashing And Pruning

QECL- Quantitative Extended Concept Lattice

RAAT- Reduced Apriori Algorithm With Tag

IAA- Improved Apriori Algorithm

AMS- Algorithm for Mining Stronger rules

AMLS- Algorithm for Mining Less Strong rules

SOT- Size Of Transaction

7.2 LIST OF PUBLICATIONS

1) A paper entitled “ Survey on various improvesd Apriori Algorithms” has been published in International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE) in November 2014, Volume 3, Issue 11.

2) A paper entitled “ To analyze the performance of Enhanced Apriori’s algorithm for association rule generation in Data Mining” has been accepted in International Journal of Applied Engineering Research (IJAER) in April 2015 which has SCOPUS indexing.(Paper code: 35205)

3) A paper entitled “ To Reduce Processing time to generate frequent itemset using Laplace equation in Apriori algorithm” has been accepted in International Conference on Advances in Applied Engineering and Technology (ICAAET)-2015. Conference Proceeding will be published in SCOPUS indexed journal International Journal of Applied Engineering Research (IJAER). (Paper Id: 1025)