Interim Report

Implementation of a Test Tool for Privacy Protection Algorithms

Author: Matthew Connop

Supervisor: Dr Jianhua Shao

Moderator: Prof. Stephen Hurley

CM0343 40 Credits

Cardiff School of Computer Science and Informatics

Cardiff University

December 2011

Abstract

Because privacy protection algorithms are important in keeping personally identifiable data secure, it is essential to be able to compare and choose the best algorithm suited for the protection of that data. This report specifies the requirements and background information needed to produce a suitable toolkit which can be used to compare algorithms. This study discusses a prototype developed to assist in the understanding of the task and problems that arose during the project.

Table of Contents

Table of Figures

Acknowledgments

I would like to thank Dr J Shao for taking his time to describe the problem in great detail and help me understand the very difficult aspects of the privacy protection world, also Dr G Loukides and Huong Ong for allowing me to use their algorithms and help in getting them to work.

1 Introduction

The project and following report is concerned with developing a java based toolkit so a user can measure and then compare privacy protection algorithms run on sets of data.

The basics of this application are to allow the user to select a dataset(s) as a control and a selection of algorithms to test; these will be the inputs for the system. The algorithm(s) will then process the dataset(s) and produce output, the user can then review the results with metric tools, allowing the algorithms performance to be compared.

The beneficiaries of this system will be researchers in the area of privacy protection; this is the study of ways to protect personally identifiable information in collections of data, for example medical data.

This is a wide field of research covering many aspects; the larger context of the problem includes many dataset formats, performance metrics and algorithms to make data anonymous and useful.

The building of the toolkit will take an iterative approach so a robust piece of software can be developed. It will be just concerned with the measuring of algorithm outputs and not the actual development of the algorithms; these will be supplied or sought after for the project.

1.1 Aims and Objectives

The overall aim of this project is to allow researchers to compare privacy protection algorithms specifically

- Improve the way algorithms are compared
- Improve the way datasets are used to prove one algorithms worth over another

The main objectives for the project are

- To provide a test bed for the algorithms
- To display results for easier interpretation of comparisons
- To learn about privacy protection algorithms
- To learn about ways in which to measure the algorithms performance
- To find suitable ways to import and manipulate datasets
- To design and build a suitable toolkit and API

1.2 Project Scope

This project will cover implementing a tool suite for users to compare the performance of privacy protection algorithms. It will include a graphical user interface for standalone use and an API (application programming interface) so the tools can be used within other applications.

The tool suite will include methods to allow a user to use existing datasets included in the system and also import new datasets, making them available for future use in testing algorithms.

As there are so many different formats and types of datasets, the system will be able to handle datasets in a .csv (comma-separated values) format. This is a widely used format for many available datasets and for that reason makes a suitable choice for the system to handle initially.

The users of the system will need the functionality to manipulate the .csv files that are available; this will come in the form of some graphical interface for removing rows and/or columns, and saving the set into the repository for later use.

This will be the same for privacy protection algorithms, the ability to use an existing algorithm or import a new algorithm. The available algorithms will be hardcoded in to the tool suite and a concept to import new algorithms will be realised within the project.

The algorithms after being run will output preserved anonymity datasets which will need to be displayed for the user to view the results. The tool suites focus will then be on allowing the user to measure the output, and compare the algorithms against each other under a variety of different criteria.

This will also come under the format of existing hard wired methods to measure the result, i.e. length of time to run, level of anonymity, and then display the results in different graphical and tabular views. A concept to import additional measures will also be realised within the project.

1.3 Outcomes

The main outcomes I would like to achieve in this project would be to enhance research into privacy protection algorithms; the system will by the use of scientific results allow researchers to prove that on the same dataset one algorithm is better than another.

Scientific papers produced in the future would be able to show results from these tests, to show how an algorithm has improved or a new algorithm is more suitable.

2 Background

Data is being collected every day, from medical records to shopping habits, this data is used by governments, organisations and individuals to data mine and gain information and insights in to the general population, for example. It can help in improving an organisations business.

To be able to share this data safely between organisations it has to be made anonymous in some way to preserve people's privacy. Privacy protection algorithms have been design to generate a level of privacy but keep the information still useful.

2.1 Problem Scope

As mentioned in the project scope datasets involved can be in varied formats. The most widely used format is a delimiter separated text file, this covers .csv files which are values separated by a comma or other formats where different delimiters have been used.

Examples of dataset types are

- Relational
- Transactional
- Networked

A relational dataset is a table of data that contains attributes (columns) and tuples (rows), the relation is the set of tuples that have the same attributes. For example, Medical data could be considered a relational dataset (see Table 2.1)

ID	Name	Age	Town	Diagnosis	Treatment
1	John	29	Cardiff	Cancer	Radiotherapy
2	Sue	34	Newport	HIV	Prophylaxis

Table 2.1 A relational dataset example

Transactional data would be describing a result of a transaction usually in the form of verbs, it has a transaction ID and will refer to one or more items or objects. For example a shopping list. (see Table 2.2)

12:84746746	Bread milk cheese lamb chops wine
47:20874017	Eggs milk chicken sherry

Table 2.2 A transaction dataset example

Networked data is a dataset describing a network of some description. For example a social network where friends are nodes and edges represent relations. It is best visualized as a graph.

These are but a few types of datasets. The datasets can be used to then data mine to find out statistics about people or places. These statistics can provide information to varied organisations be it the government, medical research companies, a supermarket or even individuals.

In order to allow these datasets to become available for the varied parties the information within them cannot give an individual's identity away or allow anyone to infer someone's identity from the data.

This is where privacy protection algorithms are used. They are a method for taking a dataset and making the data anonymous, many algorithms have been developed, some based on certain privacy models (see Table 2.3).

Privacy Model	Attack Model			
	Record Linkage	Attribute Linkage	Table Linkage	Probabilistic Attack
k-Anonymity	✓			
MultiR k-Anonymity	✓			
ϑ-Diversity	✓	✓		
Confidence Bounding		✓		
(α, k)-Anonymity		✓		
(X,Y)-Privacy	✓	✓		
(k, e)-Anonymity	✓	✓		
$(\in, m)$-Anonymity		✓		
Personalized Privacy		✓		
t-Closeness		✓		✓
δ-Presence			✓	
(c, t)-Isolation	✓			✓
$\in$-Differential Privacy			✓	✓
(d, γ)-Privacy			✓	✓
Distributional Privacy			✓	✓

Table 2.3 Privacy models (Fung, 2010, p.7)

Fung et al (2010) describes the above privacy models against attack models. This shows how some privacy models can only prevent a certain attack.

An attack is when you try to link a record in a dataset to an individual. The different privacy models try to stop these attacks as you can see in fig 2.3 they do not cover all attack models, privacy models can be split into two main categories.

First is when the attacker knows someone's identity, they may try to find out if the target is in the dataset (table linkage) or if they know that the targets record is in that dataset then they could be identified through record or attribute linkage.

Second is the probabilistic attack where it is assumed the attacker has no additional information but after viewing the dataset the attacker's knowledge or beliefs will change.

The *k*-anonymity privacy model attempts to reduce the number of individual identifiers (i.e. sex, age), it focuses on making sure that for every identifier value there is another k-1 records with the same identifier, so you could distribute age into categories, i.e. [21-25], [26-30]. For more information Fung et al further describe the other privacy models.

One problem with these algorithms is that by increasing the privacy the information gained from the processed datasets may be reduce.

This is where the measurements can help in evaluating an algorithms worth. Many metric methods have been devised, but one metric may be suitable to measure algorithm A but not algorithm B for example.

An information metric will check the similarity between the original dataset and the anonymous dataset, this method will give a penalty to an algorithm for every item that has been generalised i.e. if five entries of age have been changed to a larger set ([21-25]) it will incur 5 penalty points.

This is a form of general purpose metric of which there is also ILoss and discernability metrics. Special purpose metrics and trade off metrics can also be used to measure an algorithm; these are discussed in more detail in Appendix A

As you can see the problem encompasses many different aspects and the project will cover a small subset of these.

2.2 Similar Systems

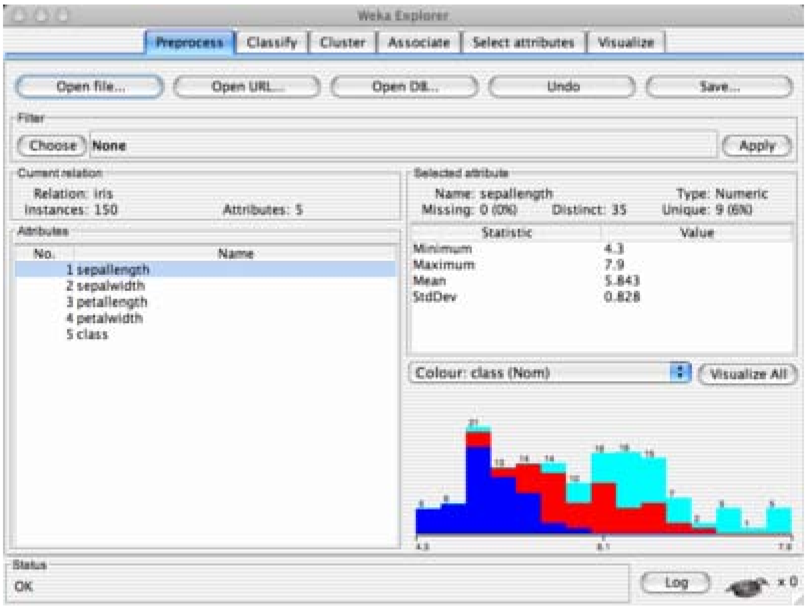
There are a few similar systems available to test privacy protection algorithms, the most prominent of these being the WEKA toolkit.

WEKA Toolkit

This is a standalone application that can be used to preserve anonymity in data sets, and also a java API that can be called from your own java program.

It contains a collection of machine learning algorithms to carry out data mining tasks. Additionally it has many tools to help with data pre-processing, classification, regression, clustering, association rules, and visualization.

The Weka Knowledge Explorer for example contains the data pre processing interface (Fig 2.1), this is the initial entry point for loading datasets and viewing them via filters and attributes.



Filter: None

Relation: iris
Instances: 150
Attributes: 5

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Status: OK

Figure 2.1 WEKA Data Pre processing interface

There are many other features in WEKA which are available to view in Appendix B.

The Cornell Anonymization Toolkit

This toolkit interactively anonymizes published datasets and will limit identification disclosure of records under various attacker models. This toolkit also includes methods for record manipulation, data generalisation and visualisation of data risks and utility as seen in Fig 2.2.

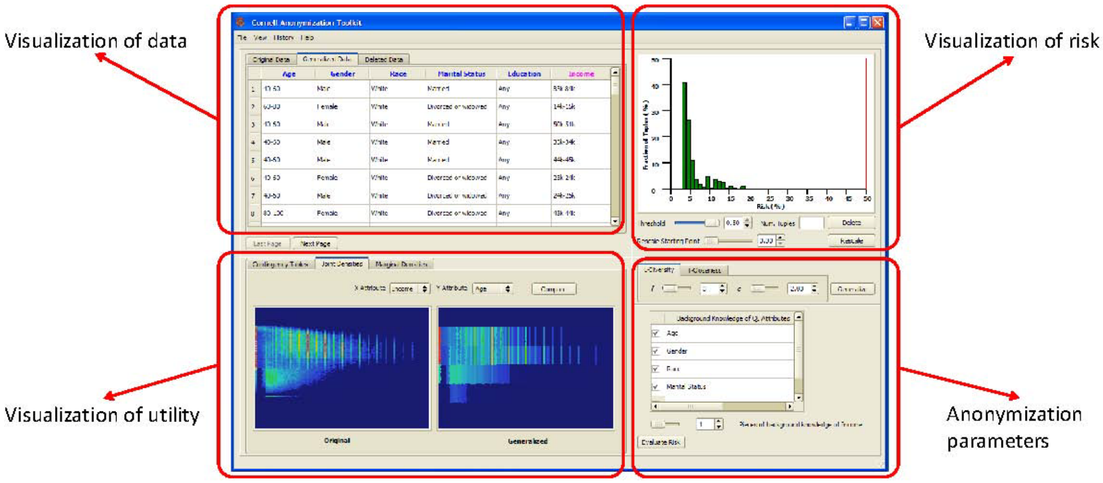


Figure 2.2 Cornell Anonymisation Toolkit interface (CAT, 2011)

UTD Anonymization ToolBox

This tool suite contains 6 different algorithms for use, allows anonymization of datasets and is based on the WEKA toolkit described earlier. It currently only allows unstructured text files as an input. The algorithm parameters have to be placed in an xml file which the toolkit reads and then runs on the dataset input it then produces a dataset in the same format as the input.

While these systems work with privacy protection algorithms they do not offer the ability to compare algorithms in a scientific way.

2.3 Project Justification

This project and tool suite will have common properties to some aspects of the similar systems. WEKA has a data importing and pre-processing which is an integral part of the project tool suite and is needed to ensure that algorithms can be tested fairly on the same datasets.

All the systems have the ability to change parameters for the algorithms i.e. setting the k value if it is a k-anonymization algorithm. This would be a highly useful feature.

Where they differ though is the ability to use your own algorithms and measures in which to test on various datasets and results. This will allow the user to compare algorithms instead of just viewing the results of one dataset run on one algorithm.

This usage can allow researchers to publish results of algorithms they have created. It will allow a way to see if an algorithm has been improved and fairly say which method is more suitable for which (type of) dataset.

2.4 Research Question Problem

Aim:
The aim of this project is to develop a software package and API for the ability to store and retrieve sets of databases, algorithms and measures for use by researchers so they can easily compare algorithms performances.

Research question:
In order to demonstrate the achievement of the stated aim, this project will evaluate software currently in use to identify desirable features, define appropriate performance metrics, determine how to compare algorithms, integrate methods to pre-process datasets, design a method to integrate new algorithms and measures, develop a user interface and API so algorithms can be judged, and implement the toolkit in a usable and robust software package.

3 Approach

The planning approach in this project took the form of identifying key requirements and outcomes that should be achieved by the project and system. This required learning about why such a system is needed, how algorithm performance can be measured, what is currently available and how the tool suite will differ. This defined an overall goal that the project could work towards.

Once the goal was identified the rest of the project took on a more agile approach to development, so while the tool suite is being developed in its iterative cycles, any problems that occur can be addressed and solved.

As the tool will be changing and progressing iteratively, the design of user interfaces and functionality will naturally change to suit the requirements and reach the goals defined. After each iterative cycle, feedback will be received by end users so the next cycle can improve on the last.

The initial main functionality of adding and manipulating datasets will be the first thing to be included in the tool suite, followed by using external algorithms to process the said datasets and finally allow the user to compare the results of these by displaying the metrics in an informative and useful way.

3.1 Specification and Design

Figure 3.1 below is a use case diagram and was used to identify a key set of requirements that the system should adhere to, assuming that during the iterative cycles the requirements will not change.

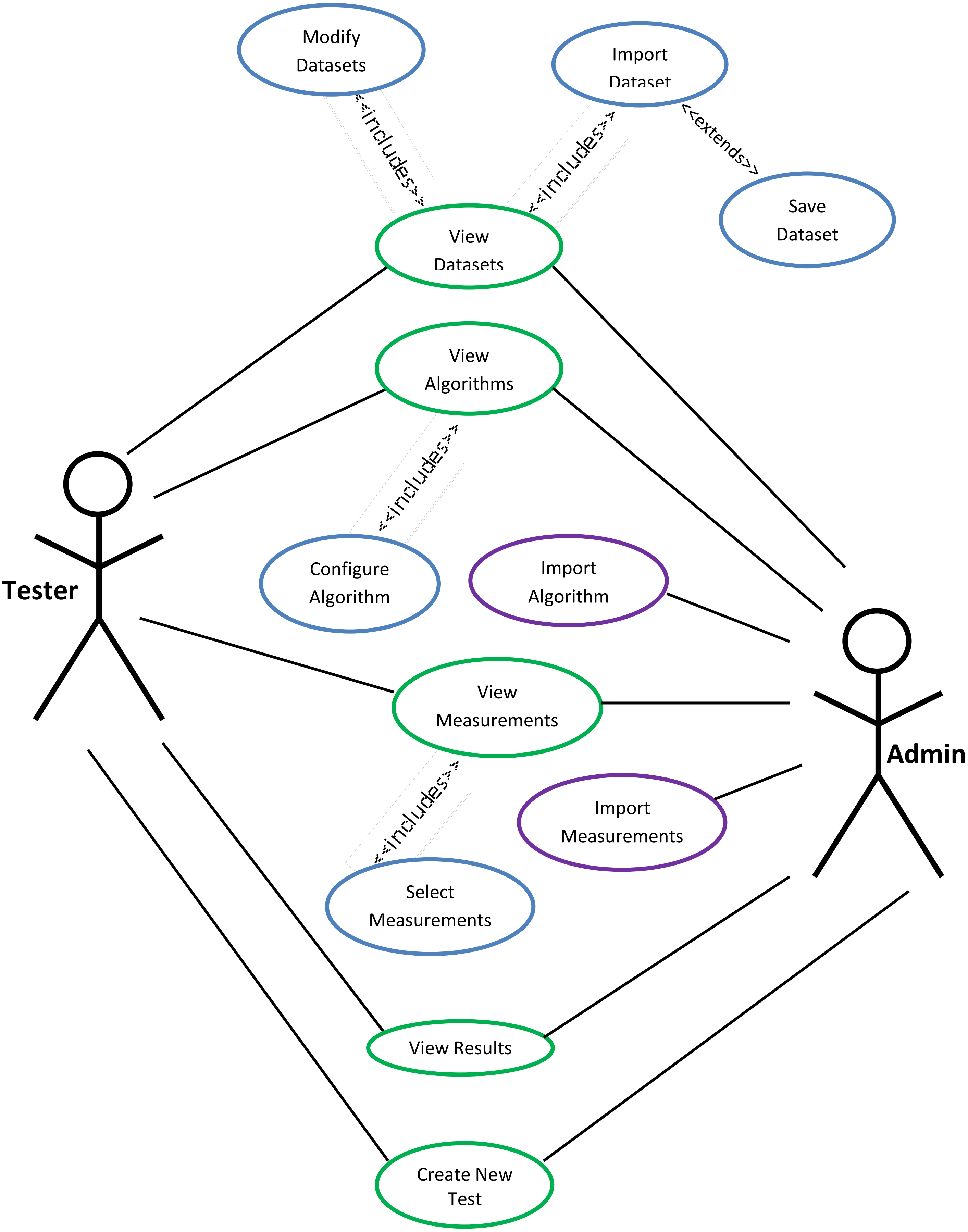


Figure 3.1 Use Case Diagram

3.1.1 Requirements

The system shall allow a user to create a new test.

The system shall allow a user to choose from a selection of datasets.

The system shall allow a user to choose from a selection of algorithms.

The system shall allow a user to review the results according to a selection of measurements.

The system shall allow a user to specify algorithm configuration.

The system shall allow manipulation of datasets.

The system shall allow a user to import more datasets.

The system shall make any imported datasets available for future tests.

The system shall produce a set of metrics from the test.

The system design shall implement a method for admin to import new algorithms for testing.

The system design shall implement a method for admin to import new measurements to review results.

The system should contain a suitable API for use in other systems.

3.1.1 Storyboards

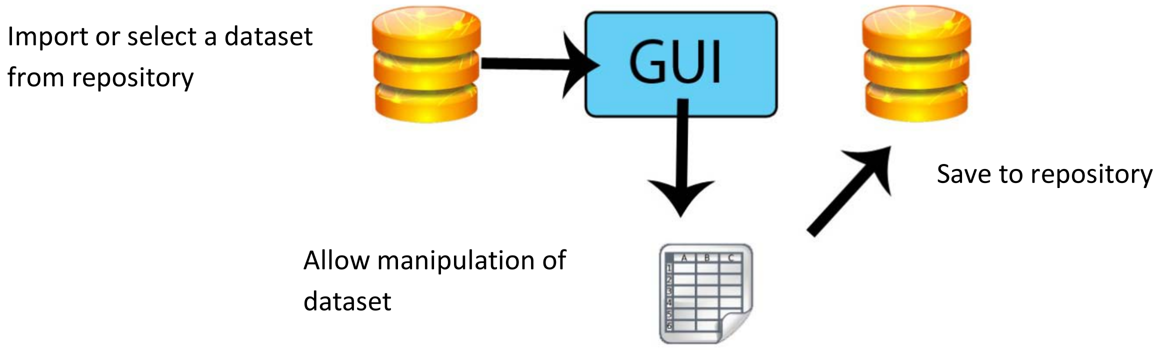

Import or select a dataset from repository
GUI
Save to repository
Allow manipulation of dataset

Figure 3.2 Database Manipulation Storyboard

This storyboard (fig 3.2) shows the basics of how the system will deal with datasets

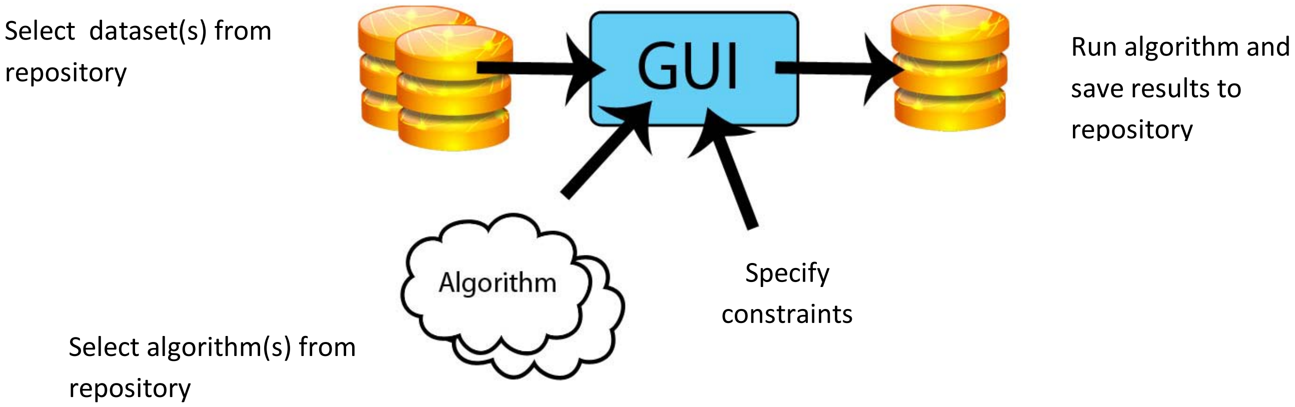

Select dataset(s) from repository
GUI
Run algorithm and save results to repository
Algorithm
Specify constraints
Select algorithm(s) from repository

Figure 3.3 Test Storyboard

Figure 3.3 shows how a test will work, what the inputs will be and the outputs

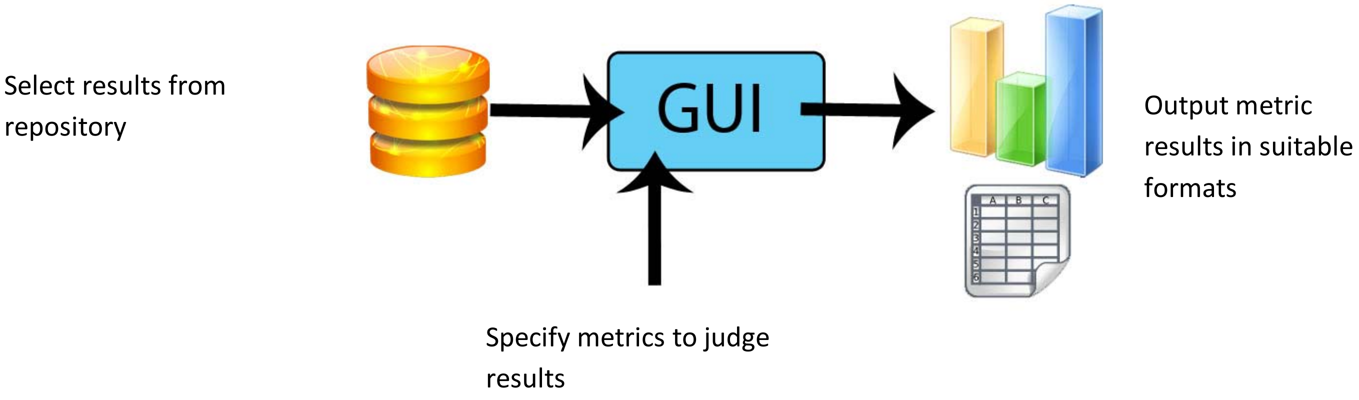

Select results from repository
GUI
Output metric results in suitable formats
Specify metrics to judge results

Figure 3.4 Comparison viewing storyboard

Figure 3.4 demonstrates how the user will view the results of the test by specifying what metrics they would like to see i.e. general purpose metric discussed in the problem scope.

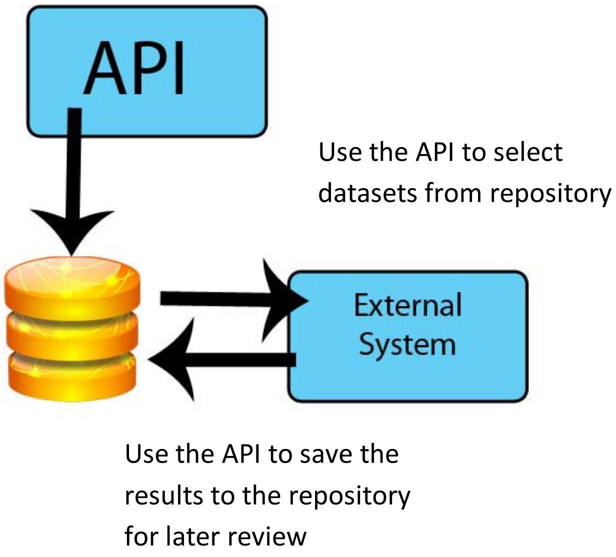

API
Use the API to select datasets from repository
External System
Use the API to save the results to the repository for later review

Figure 3.5 API example storyboard

Figure 3.5 shows how a user could use the API to access the repository for datasets and save the results back into the repository. They can then use the system GUI to review the results.

3.2 Prototype

For the interim report and the approach that the project is taking a prototype has been developed. Figure 3.6 shows the prototype interface, it is a very basic interface to get the project started and to see how to interact with the elements of the system that need to be built.

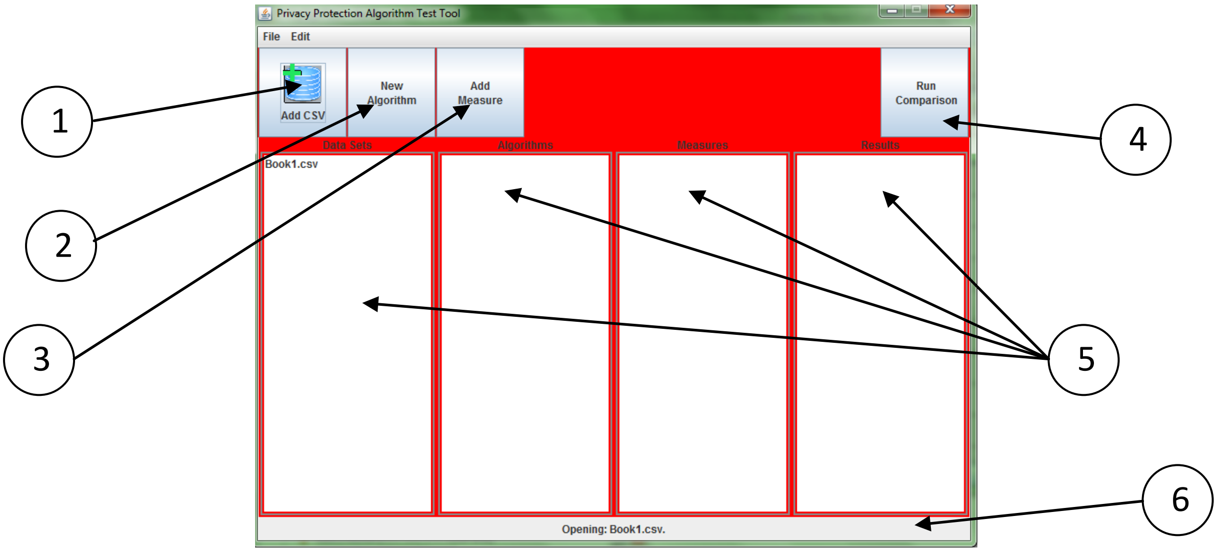


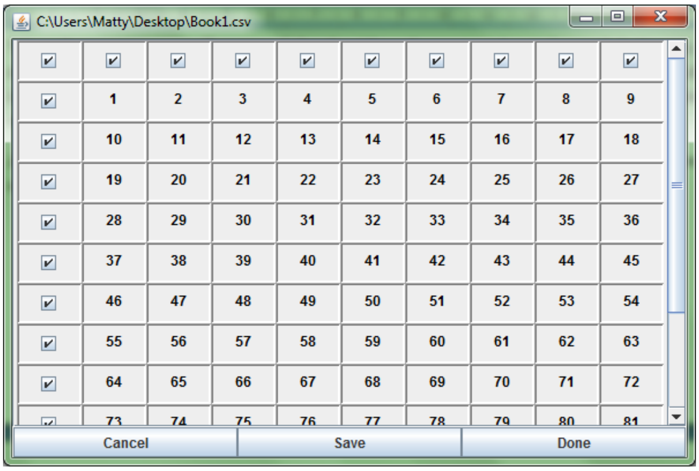
Figure 3.6 Prototype interface

Pressing the add csv button (1) will open up a file dialog that allows the user to choose a dataset from the computer. Add algorithm button (2) brings up a list of available algorithms; at this moment only one has been imported into the system. Add measure button (3) brings up a list of available measures for the user to choose from. Run comparison Button (4) will run the comparison/test.

When a user has selected their choices they are displayed in the list panes (5) to view before the test is started and the bar at (6) is an information bar giving feedback on what the system is currently doing.

When the prototype was first started the algorithms that were going to be used dealt with a relational dataset hence the .csv format, so the prototype uses a package called opencsv, which is a helper API to handle a csv file, it is a reader and writer and has no functionality to display or manipulate a .csv file.

A csv display tool has been written which allows the user to view a csv file and remove rows or columns as seen in Figure 3.7.



☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	1	2	3	4	5	6	7	8	9
☑	10	11	12	13	14	15	16	17	18
☑	19	20	21	22	23	24	25	26	27
☑	28	29	30	31	32	33	34	35	36
☑	37	38	39	40	41	42	43	44	45
☑	46	47	48	49	50	51	52	53	54
☑	55	56	57	58	59	60	61	62	63
☑	64	65	66	67	68	69	70	71	72
☑	73	74	75	76	77	78	79	80	81

Figure 3.7 Csv editor tool

This is a simple frame where rows and columns can be selected to suit the user's requirements and the file can then be saved and added to the list of datasets.

As the prototype progressed it was time to implement an algorithm, unfortunately the algorithms available to use dealt with transactional data, for which the input and output is a text file with varied delimiters and unstructured rows.

This brought additional problems, one being that the transactional data is not always comma separated or even distributed into even columns. A possible solution could be to use the openoffice.org API to show the different datasets and allow the user to manipulate the files through that.

The second problem is that in order for algorithms to be imported into the system they need to be compiled and the program needs to know the entry point into the algorithm, what additional parameters are needed and the output. A solution to this will hopefully be realised over the coming months.

So the prototype just simply allows a user to add and manipulate a csv file, the run comparison will not work with a csv file and needs a special set of transaction data to work, which is available.

There are no metric capabilities within the system at the moment and the prototype will need to be radically changed to be able to do comparisons. The user will need to import the data results from any algorithm outside the system for it to be measured; this will require the use of an API in which the external algorithm knows where to save the file.

A different interface will be needed for this where results can be scanned using a set of tools and metrics after the algorithm has been run.

4 Conclusion

In researching this area I have learnt a lot about the study of privacy protection algorithms and although this test tool does not require me to create a privacy algorithm, it has been very interesting. The problem area is a large one and the tool will cover just a small fraction of this but will hopefully be a good base to continue the tool past the project and into a proper research setting.

I have achieved the majority of the aims discussed in the initial plan; the prototype does not measure the results due to the change in format the available algorithm used, but these problems encouraged me to find solutions which will be implemented in the next stage of the project.

The more agile method that has been used greatly differs from past projects and when trying to create a timeplan for the initial plan it was very hard to envision how my time would be split up over the year and was very linear in approach. As the project has progressed the need to overlap tasks became obvious to create cycles of development and has been reflected in an updated timeplan. see Appendix C.

For future work I will find a more suitable method to allow the user to manipulate different types of data sets, not just relational data, then the current algorithm at my disposal can be used, and methods to measure its performance can be tested. I also intend to apply the tools for the API so they can be called from any program using the library.

Appendix A: Privacy Metrics

Age	Name	Diagnosis	Treatment
23	John	Bowel Cancer	Radiotherapy
32	Sue	Lung Cancer	Chemotherapy
33	Mike	Leukaemia	Bone Marrow replacement
40	Alan	Throat Cancer	Chemotherapy
40	Mary	Brain Tumour	N/A

Table A-1 example medical data

General purpose metrics:

ILoss

Iloss is a metric that generalises values and gives penalty points to loss of information. In table A-1 above there are 4 separate values for age, if we were to generalize these into [21-25], [31-35] and [36-40]. Then

$$\text{ILoss([31-35])} = \frac{\text{Number of elements in[31-35]} - 1}{\text{Number of values in age attribute}}$$

$$= \frac{2-1}{4} = \mathbf{0.25}$$

Discernibility

The discernibility metric still works by giving penalty points, but instead of giving them for generalization like ILoss, it works by giving penalty points to the number of indistinguishable records in the generalization. So the discernibility metric for [31-35] would be 2 in Table A-1.

Special purpose metrics

A special purpose metric is one used for data that when published its purpose is known in advance. So if an attribute in a table like age in table A-1 is important for the purposes of the data miner, then any generalization of this would incur a penalty as described by that special metrics rules.

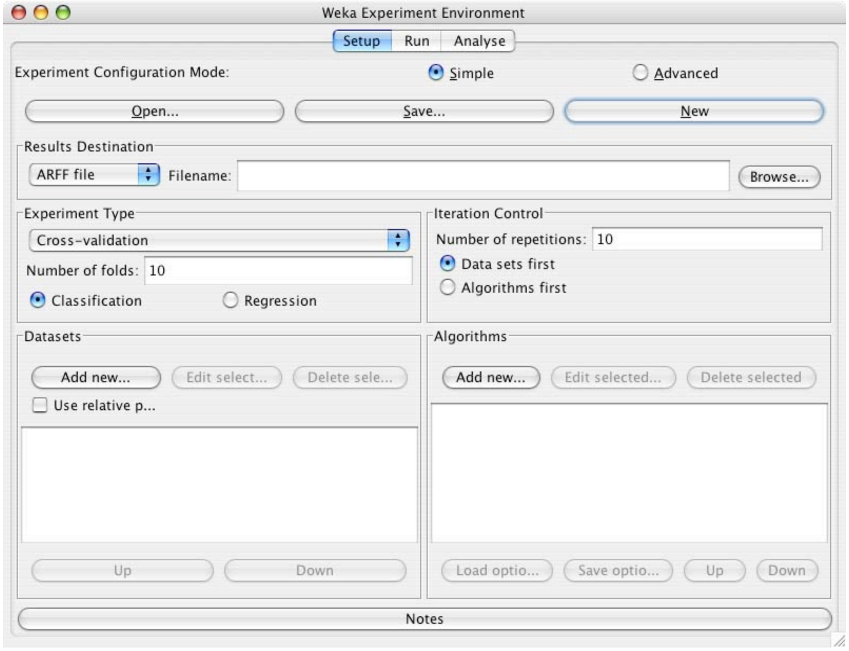
Trade off metrics

Trade off metrics measure the algorithms specialisation at operations. It is based on losing points for information loss and gaining points for privacy for each attribute entry that is placed in a generalization category.

Appendix B: WEKA toolkit

The WEKA toolkit uses a special format designed for their purpose called ARFF files, where attributes are defined at the top of the file and the date conforming to these attributes below, it is a bit like XML.

As mentioned earlier the WEKA explorer deals with pre-processing the data and contains some very useful tools for visualizing the data including tress and graphs. It also has an experimenter tool. (Figure B-1)



Weka Experiment Environment

Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination

ARFF file Filename:

Experiment Type

Cross-validation

Number of folds: 10

● Classification ○ Regression

Iteration Control

Number of repetitions: 10

● Data sets first

○ Algorithms first

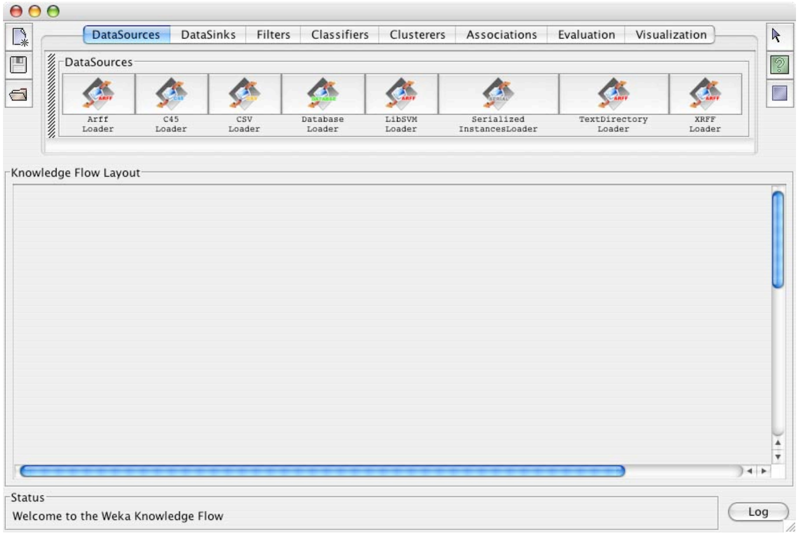
Datasets

□ Use relative p...

Figure B-1 WEKA Experiment interface

The experimenter allows the comparison of the available learning schemes performance and can evaluate the learning curve and cross validation.

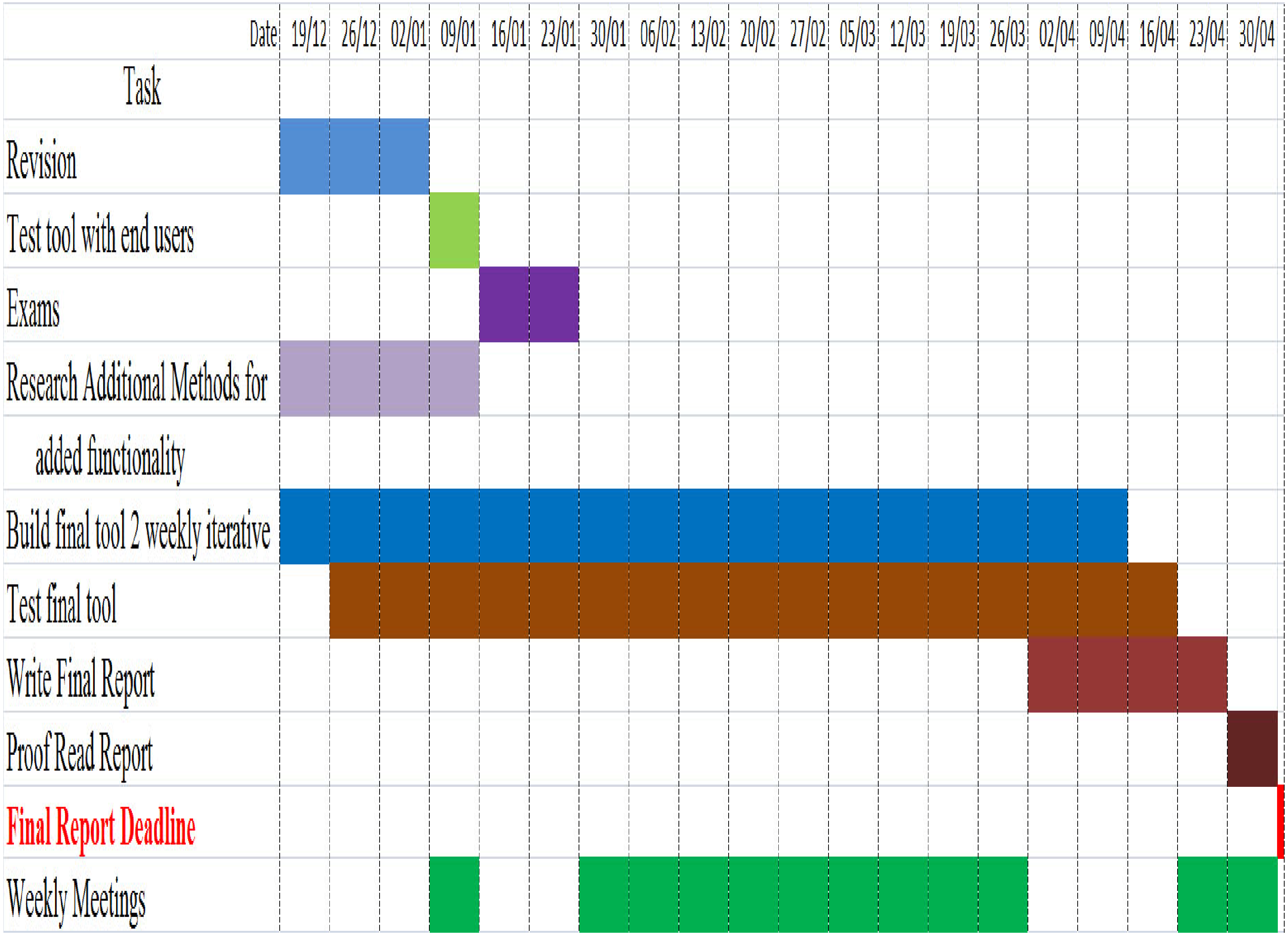
The final tool in the WEKA tool kit is called the Knowledge Flow GUI; this is a graphical interface for running machine learning experiments. (Fig B-2) Data Sources can be connected in a graphical interface and applied through filters and classifiers which can then be visualised.



Welcome to the Weka Knowledge Flow

Figure B-25 WEKA Knowledge Flow GUI

Appendix C: Updated Timeplan



Date	19/12	26/12	02/01	09/01	16/01	23/01	30/01	06/02	13/02	20/02	27/02	05/03	12/03	19/03	26/03	02/04	09/04	16/04	23/04	30/04
Task																				
Revision	■	■	■																	
Test tool with end users				■																
Exams					■	■														
Research Additional Methods for added functionality	■	■	■	■																
Build final tool 2 weekly iterative	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Test final tool		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Write Final Report																■	■	■	■	
Proof Read Report																				■
Final Report Deadline																				🟥
Weekly Meetings				■			■	■	■	■	■	■	■	■	■				■	■

Figure C-1 Timeplan

References

CAT. 2011. *CAT Anonymization Interface* . [image online]
Available at: http://switch.dl.sourceforge.net/project/anony-toolkit/Documents/cat-mannual-1.0.PDF [Accessed 10 December 2011].

Cornell Anonymization Toolkit Available at: http://sourceforge.net/projects/anony-toolkit/ [Accessed 21 October 2011].

Fung, C et al. 2010. *Privacy-preserving data publishing: A survey of recent developments*, ACM Computing Surveys, 42(4) Article 14. Available through: ACM Digital Library

UTD Anonymization ToolBox, Available at: http://cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php [Accessed 4 November 2011].

WEKA Toolkit, Available at: http://www.cs.waikato.ac.nz/ml/weka/
[Accessed 15 October 2011].