ASSOCIATIVE CLASSIFICATION TECHNIQUES FOR E-BANKING PHISHING WEBSITES

A PROJECT REPORT

Submitted by

AARTHI.N, Reg.No: 0710108001

SARANYA.R, Reg.No: 0710108044

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

Autonomous institution Affiliated to Anna University of Technology, Coimbatore.

APRIL, 2011

BONAFIDE CERTIFICATE

Certified that this project report entitled "**Associative Classification Techniques for Predicting e-Banking Phishing Websites** " is the bonafide work of Aarthi.N and Saranya.R who carried out the research under my supervision. Certified also, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Head of the Department | **Signature of the Guide**

(Mrs.P.Devaki, M.E.,) | **(Mrs.R.Kalaiselvi, M.E.,)**

The candidate with University **Register Nos. 0710108001 and 0710108044** were examined by us in the project viva-voce examination held on 20·4·11

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled " **Associative Classification Techniques for Predicting e-Banking Phishing Websites**" is a record of original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any Institutions, for fulfillment of the requirement of the course study.

The report is submitted in partial fulfillment of the requirement for the award of the Degree of Bachelor of Computer Science and Engineering of Anna University, Coimbatore.

Place: Coimbatore

Date: 19|4|11

Aarthi N.

(AARTHI.N)

Saranya R.

(SARANYA.R)

ACKNOWLEDGEMENT

We express our profound gratitude to our chairman Padmabhusan **Arutselver Dr.N.Mahalingam, B.sc, F.I.E.**, and Correspondent Shri. **Balasubramanian, M.com., M.B.A.**, for given us this opportunity to embark on this project.

We extend our sincere thanks to our director **Dr.J.Shanmugam**, and our principal, **Dr.S.Ramachandran Ph.D.**, Kumaraguru College of Technology, Coimbatore, for being a constant source of inspiration and providing us with the necessary facilities and infrastructure to work on this project.

We are intended to express our heartiest thanks to **Mrs.P.Devaki, M.E.**, Project coordinator, Head of the Department of Computer Science & Engineering, for her valuable guidance and useful suggestions during the course of this project.

We express our deep sense of gratitude and gratefulness to our guide **Mrs.R.Kalaiselvi, M.E.**, Assistant Professor in Computer Science and Engineering, for her supervision, enduring patience, active involvement and guidance.

We would like to convey our honest thanks to all faculty members of the Department for their enthusiasm and wealth of experience from which we have greatly benefited. We also thank our friends and family who helped us to complete this project fruitfully.

TABLE OF CONTENTS

Abstract:

We present a novel approach to overcome the difficulty and complexity in detecting and predicting e-banking phishing website. We proposed an intelligent resilient and effective model that is based on using association and classification data mining algorithms. These algorithms were used to characterize and identify all the factors and rules in order to classify the phishing website and the relationship that correlate them with each other. We implemented PART classification algorithm and techniques to extract the phishing training data sets criteria to classify their legitimacy. The rules generated from the associative classification model showed the relationship between some important characteristics like URL and domain identity, security and encryption criteria in the final phishing detection rate.

CHAPTER 1

1.1 Phishing

Phishing is an e-mail fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients. Typically, the messages appear to come from well known and trustworthy web sites. Web sites that are frequently spoofed by phishers include PayPal, eBay, MSN, Yahoo, BestBuy, and America Online. A phishing expedition, like the fishing expedition it's named for, is a speculative venture: the phisher puts the lure hoping to fool at least a few of the prey that encounter the bait.

Phishers use a number of different social engineering and e-mail spoofing ploys to try to trick their victims. In one fairly typical case before the Federal Trade Commission (FTC), a 17-year-old male sent out messages purporting to be from America Online that said there had been a billing problem with recipients' AOL accounts. The perpetrator's e-mail used AOL logos and contained legitimate links. If recipients clicked on the "AOL Billing Center" link, however, they were taken to a spoofed AOL Web page that asked for personal information, including credit card numbers, personal identification numbers (PINs), social security numbers, banking numbers, and passwords. This information was used for identity theft.

1.2 Data Mining

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

1.3 Objective

The objective and the motivation behind this study is to create a resilient and effective method that uses data mining algorithms and tools to detect e-banking phishing websites in an artificial intelligence technique. Associative and classification algorithms can be very useful in predicting phishing websites.

CHAPTER 2

2. SYSTEM ANALYSIS

2.1 Existing System

1. The approach described here is to apply data mining algorithms to assess e-banking phishing website risk on the 27 characteristics and factors which stamp the forged website.
2. Associative and classification algorithms can be very useful in predicting phishing websites.
3. It can give us answers about what are the most important e-banking phishing website characteristics and indicators and how they relate with each other.
4. The choice of PART algorithm is based on the fact that it combines both approaches to generate a set of rules.
5. Associative classifiers produce more accurate classification models and rules than traditional classification algorithms.

2.2 Proposed System

In the proposed system, we present novel approach to overcome the 'fuzziness' in the e-banking phishing website assessment and propose an intelligent resilient and effective model for detecting e-banking phishing websites. There is a significant relation between the two phishing website criteria's *(URL & Domain Identity)* and *(Security & Encryption)* for identifying e-banking phishing website. Also we found insignificant trivial influence of the *(Page Style & Content)* criteria along with *(Social Human Factor)* criteria for identifying e-banking phishing websites.

2.3 Feasibility Study

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project.

2.3.1 Economical Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

2.3.2 Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.

- Does management support the project?
- Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.
- Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.

- Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system effect the customers in considerable way?

2.3.3 Technical Feasibility

In technical feasibility the following issues are taken into consideration.

- Whether the required technology is available or not
- Whether the required resources are available – Manpower - programmers, testers & debuggers- Software and hardware

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

2.4 LITERATURE SURVEY

2.4.1 Introduction

"Phishing" is the term for an e-mail scam that spoofs legitimate companies in an attempt to defraud people of personal information such as logins, passwords, credit card numbers, bank account information and social security numbers. For example, an e-mail may appear to come from PayPal claiming that the recipient's account information must be verified because it may have been compromised by a third party. However, when the recipient provides the account information for verification, the information is really sent to a phisher, who is then able to access the person's account. The term phishing was coined because the phishers are "fishing" for personal information. Phishing e-mails are sent to both consumers and companies, trying to gain either personal information from an individual or confidential information about an enterprise. In phishing e-mail messages, the senders must gain the trust of the recipients to convince them to divulge information. The phishers attempt to gain credibility through mimicking or "spoofing" a legitimate company through methods such as using the same logos and color scheme, changing the "from" field to appear to come from someone in the spoofed company, and adding some legitimate links in the e-mail.

One approach is to stop phishing at the email level, since most current phishing attacks use broadcast email (spam) to lure victims to a phishing website. Another approach is to use security toolbars. The phishing filter in IE7 is a toolbar approach with more features such as blocking the user's activity with a detected phishing site. A third approach is to visually differentiate the phishing sites from the spoofed legitimate sites. Dynamic Security Skins proposes to use a randomly generated visual hash to customize the browser window or web form elements to indicate the successfully authenticated sites. A fourth approach is two-factor authentication, which ensures that the user not only knows a secret but also presents a security token . Many industrial antiphishing products use toolbars in web browsers, but some researchers have shown that security tool bars don't effectively prevent phishing attacks. Another approach is to employ certification, e.g., Microsoft spam privacy. A variant of web credential is to use a database or list published by a trusted party, where known phishing web sites are blacklisted. The weaknesses of this approach are its poor scalability and its timeliness. The newest version of Microsoft's Internet Explorer supports Extended Validation (EV) certificates, coloring the URL bar green and displaying the name of the company. However, a recent study found that EV certificates did not make users less fall for phishing attacks.

2.4.2 Filtering approaches for phishing e-mail

Phishing emails usually contain a message from a credible looking source requesting a user to click a link to a website where she/he is asked to enter a password or other confidential information. Most phishing emails aim at withdrawing money from financial institutions or getting access to private information. Phishing has increased enormously over the last years and is a serious threat to global security and economy. There are a number of possible countermeasures to phishing. These ranges from communication-oriented approaches like authentication protocols over blacklisting to content-based filtering approaches.

We argue that the first two approaches are currently not broadly implemented or exhibit deficits. Therefore content-based phishing filters are necessary and widely used to increase communication security. A number of features are extracted capturing the content and structural properties of the email. Subsequently a statistical classifier is trained using these features on a training set of emails labeled as ham (legitimate), spam or phishing. This classifier may then be applied to an email stream to estimate the classes of new incoming emails.

In this paper we describe a number of novel features that are particularly well-suited to identify phishing emails. These include statistical models for the low-dimensional descriptions of email topics, sequential analysis of email text and external links, and the detection of embedded logos as well as indicators for hidden salting. Hidden salting is the intentional addition or distortion of content not perceivable by the reader. For empirical evaluation we have obtained a large realistic corpus of emails prelabeled as spam, phishing, and ham (legitimate). In experiments our methods outperform other published approaches for classifying phishing emails.

2.4.3 Antiphish-machine learning for phishing detection

The antiphish is a prototype system that is highly accurate in the detection of phishing email messages. In the first phase of the antiphish project a machine learning prototype called Antiphish Filter System (APS) was developed and evaluated on public benchmark data. It combines a number of novel features that are particularly well-suited to identify phishing emails. In particular we investigated Latent Dirichlet Allocation (LDA) topic models to capture words that frequently co-occur in email messages. We developed a special version called the latent Class-Topic Model (CLTOM), which is an extension of Latent Dirichlet Allocation (LDA) in such a way that it incorporates category information of emails during the model inference for topic extraction [BCP+08]. In addition we derived an optimized version of Dynamic Markov Chain (DMC) models, which generates far smaller models without sacrificing performance [BCP+08]. We combined these features with other standard and image features and trained a classifier using feature selection. In experiments our methods increase the f-value for classifying phishing emails on public benchmark data from 97.6% [FST07] to 99.5% [BDG+09]. Common tricks of spammers known as message salting are the inclusion of random strings and diverse combinatorial variations of spacing, word spelling, word order, etc. Some salting techniques called hidden salting cause messages to visually appear the same to the human eye, even if the machine-readable forms are very different. We rendered the email image to detect the appearance and overlap of characters. Using this evidence we developed specific classifiers for identifying hidden salting of new types using outlier detection methods. We applied the APS to real-world spam and phishing detection. In the first field experiment APS was applied to the real email stream at an Internet Service Provider. Starting with an initial labeled sample to estimate a starting classifier for ham vs. non-ham (spam + phishing) we applied active learning to select new emails for classification. These selected emails have to be labeled, which can be done by volunteer customers. The APS achieved good results as standalone filter with 0.34% false positives (ham classified as non-ham) and 7.1% false negatives. The combination with a commercial spam filter (which evaluates blacklists) could improve the performance of both filters yielding 0.33% false positives and 5.4% false negatives. The second field experiment was devoted to the analysis of known spam and phishing emails from a honey pot network. The task is to separate phishing emails from spam emails in a constant message stream. This is especially important as most phishing scams exist only a few hours. To update commercial phishing filters immediately it is important to capture phishing emails which are not covered by the current phishing signatures. From the emails new rules may be created and forwarded to customers. In a sliding window approach (4 weeks training, one week prediction) the emails were monitored for 6 months. The results show that the APS has a good performance with low errors: 0.18 % spam was classified as phishing, 4.9% phishing classified as spam. More importantly APS detected a large number of emails which were not captured by current filtering rules. Hence APS permits prioritization of phishing-filter updating, which is most important because of the high damage caused by phishing emails.

2.4.4 Bayesian phish filtering

Today most anti-spam and e-mail security products have some type of phishing protection. Nearly all of these products rely principally on some form of URL detection and similar techniques noted previously or by using their spam filtering techniques. As we have noted, using anti-spam techniques on a phishing e-mail is insufficient because spam and phishing e-mails are truly different in their purpose, construction and social engineering techniques. Further, Bayesian phishing detection techniques can dramatically reduce the chances of anyone in your organization from being exposed to a phishing e-mail.

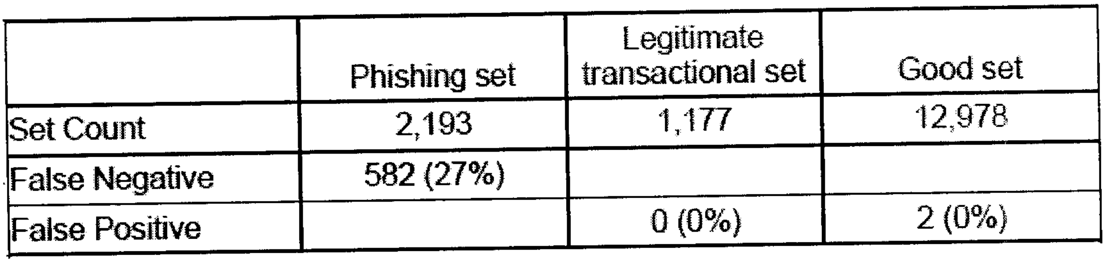
Bayesian spam filtering has a well established history as an anti-spam weapon. However, these filters are less than effective at identifying phishing emails. Spam emails are generally a sales pitch aimed at promoting a product or service. While phishing emails are designed to look like legitimate transactional correspondence and almost always work to disguise their true source. To accurately catch phishing emails, Bayesian filters must be specifically designed for that purpose.

Blocking phish faster

Blocking phishing e-mails through real-time black lists, reputation services and other related methods require that the phishing e-mail first be received by someone and then analyzed to ensure that the message is in fact phishing. Only then can the "phishing" designation for that e-mail be communicated to other parties (customers). In other words someone has to be exposed to the phishing threat before it can be identified and stopped. With Bayesian phish filtering, a phishing message can be determined as phishing "on-the-spot" the first time it is seen as part of the analysis done by Sonicwall Email Security and Sonicwall Anti-Spam products and services. In addition, that phishing designation can be communicated to other parties via the Sonicwall grid Network.

Testing the filter

Sonicwall trained a Bayesian phishing filter using the methods discussed above and then tested the filter for effectiveness. The test e-mails were passed through the Bayesian phishing filter and gave the following results.



	Phishing set	Legitimate transactional set	Good set
Set Count	2,193	1,177	12,978
False Negative	582 (27%)		
False Positive		0 (0%)	2 (0%)

Table – Bayesian Phishing Test Results

2.4.5 Security toolbars

The first attempt specifically designed to filter phishing attacks have taken the form of browser toolbars, such as the spoofguard and netcraft toolbars. Most toolbars are lucky to get 85% accuracy identifying phishing websites. Accuracy aside, there are both advantages disadvantages to toolbars when compared to email filtering.

The email provides the context under which the attack is delivered to the user. An email filter can see what words are used to entice the user to take action, which is currently not known to a filter operating in a browser separate from the user's e-mail client. An email filter also has access to header information, which contains not only information about who sent the message, but also information about the route the message took to reach the user. This context is not currently available in the browser with given toolbar implementations. Future work to more closely integrate a user's email environment with their browser could alleviate these problems, and would actually provide a potentially richer context in which to make a decision. There are some pieces of information available in the web browser and website itself that could help to make a more informed decision, especially if this information could be combined with the context from the initial attack vector, such as the email prompting a user to visit a given website. Toolbars usually prompt users with a dialog box, which many users will simply dismiss or misinterpret, or worse yet these warning dialogs can be intercepted by user-space malware. By filtering out phishing emails before they are ever seen by users, we avoid the risk of these warnings being dismissed by or hidden from the user. We also prevent the loss of productivity suffered by a user who has to take time to read, process, and delete these attack emails.

2.4.6 Machine-learning based approach to classification

In a general sense, we are deciding whether some communication is deceptive, i.e. whether it is designed to trick the user into believing they are communicating with a trusted source, when in reality the communication is from an attacker. We make this decision based on information from within the email or attack vector itself (an internal source), combined with information from external sources. This combination of information is then used as the input to a classifier, the result of which is a decision on whether the input contained data designed to deceive the user. With respect to email classification, we have two classes, namely the class of phishing emails, and the class of good emails. In this paper we present a collection of features that has been identified as being particularly successful at detecting phishing, given the current state of attacks. We expect that over time, as the attacks evolve, new sets of features will have to be identified combining information from both internal and external sources.

2.4.7 Features used in e-mail classification

Some spam filters use hundreds of features to detect unwanted emails. We have tested a number of different features, and present in this paper a list of ten features that are used in PILFER, which are either binary or continuous numeric features. As the nature of phishing attacks changes, additional features may become more powerful, and PILFER can easily be adapted by providing such new features to the classifier. At this point, however, we are able to obtain high accuracy with only ten features, which makes the decision boundaries less complex, and therefore both more intuitive and faster to evaluate. We explain these features in detail below. While some of these features are already implemented in spam filters (such as the presence of IP-based URLs), these features are also useful components of a phishing filter.

IP based URLs

These machines may not have DNS entries, and the simplest way to refer them is by IP address. Companies rarely link to pages by an IP-address, and so such a link in an email is a potential indication of a phishing attack. As such, anytime we see a link in an email whose host is an IP-address (such as http://192.168.0.1/paypal.cgi?fix account), we flag the email as having an IP-based URL. As phishing attacks are becoming more sophisticated, IP-based links are becoming less prevalent, with attackers purchasing domain names to point to the attack website instead. However, there are still a significant number of IP-based attacks, and therefore this is still a useful feature.

Links to domain names

Phishers are learning not to give themselves away by using IP-based URLs. Name-based attacks, in which a phisher will register a similar or otherwise legitimate-sounding domain name (such as playpal.com or paypal-update.com) are increasingly common. These domains often have a limited life, however. Phishers may register these domains with fraudulently obtained credit cards (in which case the registrar may cancel the registration), or the domain may be caught by a company hired to monitor registrations that seem suspicious. (Microsoft, for instance, watches for domain name registrations involving any of their trademarks.) As such, the phisher has an incentive to use these domain names shortly after registration.

.

Non-matching URLs

Phishers often exploit HTML emails, in which it is possible to display a link that says paypal.com but actually links to badsite.com. For this feature, all links are checked, and if the text of a link is a URL, and the HREF of the link is to a different host than the link in the text, the email is flagged with a "no matching URL" feature. Such a link looks like <a href="badsite.com"> paypal.com</a>.

2.4.8 Effective techniques to detect phishing sites

To analyze the effectiveness of anti-phishing solutions more precisely, we are interested in assessing techniques that are capable of classifying individual web pages. To qualify for our study, a technique must be capable of determining whether a page is legitimate or a phishing page, given only the URL and the page's source code. We did not consider mechanisms that aim to prevent users from visiting a phishing site (e.g., by recognizing phishing mails). Also, we did not evaluate solutions that attempt to protect sensitive user information from being leaked to the phishers (e.g., by replacing passwords with site-specific tokens, or by using novel authentication mechanisms). Currently, there are two main approaches to classify visited web pages without any additional information. The first one is based on URL blacklists. The second approach analyzes properties of the page and (sometimes) the URL to identify indications for phishing pages.

Blacklists

Blacklists hold URLs (or parts thereof) that refer to sites that are considered malicious. Whenever a browser loads a page, it queries the blacklist to determine whether the currently visited URL is on this list. If so, appropriate countermeasures can be taken. Otherwise, the page is considered legitimate. The blacklist can be stored locally at the client or hosted at a central server. Obviously, an important factor for the effectiveness of a blacklist is its coverage. The coverage indicates how many phishing pages on the Internet are included in the list. Another factor is the quality of the list. The quality indicates how many non-phishing sites are incorrectly included into the list. For each incorrect entry, the user experiences a false warning when she visits a legitimate site, undermining her trust in the usefulness and correctness of the solution. Finally, the last factor that determines the effectiveness of a blacklist-based solution is the time it takes until a phishing site is included. This is because many phishing pages are short-lived and most of the damage is done in the time span between going online and vanishing. Even when a blacklist contains many entries, it is not effective when it takes too long until new information is included or reaches the clients.

For our study, we attempted to measure the effectiveness of popular black-lists. In particular, we studied the blacklists maintained by Microsoft and Google. We believe that these blacklists are the ones that are most wide-spread, as they are used by Internet Explorer and Mozilla Firefox, respectively. Page analysis: Page analysis techniques examine properties of the web page and the URL to distinguish between phishing and legitimate sites. Page properties are typically derived from the page's HTML source. Examples of properties are the number of password fields, the number of links, or the number of unencrypted password fields (these are properties used by spoofguard). The effectiveness of page analysis approaches to identify phishing pages fundamentally depends on whether page properties exist that allow to distinguish between phishing and legitimate sites. Thus, for our study, we aimed to determine whether these properties exist, and if so, why they might be reasonable candidates to detect phishing pages.

In a first step, we defined a large number of page properties that can be extracted from the page's HTML code and the URL of the site. Then, we analyzed a set of phishing and legitimate pages, assigning concrete values to the properties for each page. Finally, using the collected data as training input, we applied machine-learning techniques to create a web page classifier. The resulting classifier is able to distinguish well between phishing and legitimate classifiers, with a very low false positive rate. This indicates that the aforementioned page properties that allow one to identify malicious pages do indeed exist, at least for current phishing pages. It seems that Microsoft has drawn a similar conclusion, as the new Internet Explorer browser also features a phishing page detection component based on page properties. This component is invoked as a second line of defense when a blacklist query returns no positive result for a visited URL.

2.4.9 Two-factor authentication

An authentication factor is a piece of information and process used to authenticate or verify the identity of a person or other entity requesting access to online resources. User authentication for most web sites and services today is accomplished by means of a single authentication factor: a password. Where a higher level of assurance is required (e.g. for access to on online banking service), a second factor is typically employed in addition to the password – hence "two factor authentication" (also called "multi factor authentication" or "strong authentication").

There are three main types of authentication factor:

- Knowledge factors – e.g. passwords, PINs;
- Possession factors – e.g. ID cards, tokens;
- Human factors (aka biometrics) – e.g. fingerprints, iris scans.

Some security practitioners argue that "true" two factor authentication requires two distinct types of factor; however, this is just a matter of semantics. There is nothing inherently less secure about using two factors of the same type.

CHAPTER 3

3. SYSTEM SPECIFICATION

3.1 Hardware Requirements

Processor	:	Pentium IV
Speed	:	Above 500 MHz
RAM capacity	:	2 GB
Hard disk drive	:	80 GB
Key Board	:	Samsung 108 keys
Mouse	:	Logitech Optical Mouse
Printer	:	DeskJet HP
Motherboard	:	Intel
Cabinet	:	ATX
Monitor	:	17" Samsung

3.2 Software Requirements

Operating System	:	Windows XP and above
Front end used	:	Java
Back End	:	SQL Server

CHAPTER 4

4 .SOFTWARE DESCRIPTION

4.1 Java-frontend

Java

Java is related to C++, which is a direct descendant of C. The trouble with C and C++ is that they are designed to be compiled for a specific target. But Java is a portable, platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments. Java can be used to create two types of programs: applications and applets. An application is a program that runs on our computer, under the operating system of that computer. An applet is an application designed to be transmitted over the Internet and executed by a Java-compatible web browser. Java is simple, secure, portable, object-oriented, robust, multithreaded, architectural-neutral, interpreted, high performance, distributed, and dynamic.

Security:

When a java comparable web server is used, the user can download applets without the fear of virus infection. Java achieves this protection by confining a java program to the java execution environment and not allowing its access to other parts of the computer.

Portability

Many types of computers and operating systems are in use throughout the world and many are connected to the internet. For programs to be dynamically downloaded to all various type of platform connected to the internet, some means of generating portable executable code is needed. The same mechanism that helps ensure security also helps create portability. Indeed, java's solution to these two problems is both elegant and efficient.

Object-Oriented

The object model in java is simple and easy to extend, while simple types, such as integers are kept as high-performance nonobjective.

Robust:

The ability to create robust programs was given a high priority in the design of java. To gain reliability, java restrict user in a few key areas, to force to find mistakes in early in program development. At the same time, java frees the user from having to worry about many of the most common causes of programming errors. Because java is strictly typed language, it checks the user code at compile time and it also checks the code at runtime.

Multithreaded

Java is designed to meet the real-world requirements of creating interactive, networked programs. To accomplish this, java supports multithreaded programming, which allows the user to write programs that do work simultaneously.

Distributed

Java is designed for distributed environment of the internet, because it handles TCP/IP protocols. The original version of java(Oak) include features for intra-address-space messaging. This allows objects on two computers to execute procedures remotely. Java has revived these interfaces in a package called Remote Method Invocation (RMI).

Dynamic

Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at runtime. This makes it possible to dynamically link code in a safe and expedient manner

4.2 SQL Server 2000-backend

Microsoft SQL Server 2000 is a full-featured Relational Database Management System (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online. Let's take a brief look at each:

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager is used to control the MSSQLServer (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServerAgent processes. An icon for this service normally resides in the system tray of machines running SQL Server. You can use service manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

Data storage

The main unit of data storage is a database, which is a collection of tables with typed columns. SQL Server supports different data types, including primary types such as *Integer*, *Float*, *Decimal*, *Char* (including character strings), *Varchar* (variable length character strings), binary (for unstructured blobs of data), *Text* (for textual data) among others. It also allows user-defined composite types (UDTs) to be defined and used. SQL Server also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). A database can also contain other objects including views, stored procedures, indexes and constraints, in addition to tables, along with a transaction log. A SQL Server database can contain a maximum of 2^{31} objects, and can span multiple OS-level files with a maximum file size of 2^{20} TB. The data in the database are stored in primary data files with an extension .mdf. Secondary data files, identified with an .ndf extension, are used to store optional metadata. Log files are identified with the .ldf extension.

Storage space allocated to a database is divided into sequentially numbered *pages*, each 8 KB in size. A *page* is the basic unit of I/O for SQL Server operations. A page is marked with a 96-byte header which stores metadata about the page including the page number, page type, free space on the page and the ID of the object that owns it. Page type defines the data contained in the page - data stored in the database, index, allocation map which holds information about how pages are allocated to tables and indexes, change map which holds information about the changes made to other pages since last backup or logging, or contain large data types such as image or text. While page is the basic unit of an I/O operation, space is actually managed in terms of an *extent* which consists of 8 pages. A database object can either span all 8 pages in an extent ("uniform extent") or share an extent with up to 7 more objects ("mixed extent"). A row in a database table cannot span more than one page, so is limited to 8 KB in size. However, if the data exceeds 8 KB and the row contains *Varchar* or *Varbinary* data, the data in those columns are moved to a new page (or possible a sequence of pages, called *Allocation unit*) and replaced with a pointer to the data.[2] For physical storage of a table, its rows are divided into a series of partitions (numbered 1 to n). The partition size is user defined; by default all rows are in a single partition. A table is split into multiple partitions in order to spread a database over a cluster. Rows in each partition are stored in either B-tree or heap structure. If the table has an associated index to allow fast retrieval of rows, the rows are stored in-order according to their index values, with a B-tree providing the index. The data is in the leaf node of the leaves, and other nodes storing the index values for the leaf data reachable from the respective nodes. If the index is non-clustered, the rows are not sorted according to the index keys. An indexed view has the same storage structure as an indexed table. A table without an index is stored in an unordered heap structure. Both heaps and B-trees can span multiple allocation units.

CHAPTER 5

5. PROJECT DESCRIPTION

5.1 Problem Definition

The efficacy of phishing attacks is diminished when users can not reliably distinguish and verify authoritative security indicators. Unfortunately, current browser and related application programs have not been carefully designed with "security usability" in mind. As a result, users have the following problems:

Users can not reliably correctly determine sender identity in email messages. The email sender address is often forged in phishing attacks. Most users do not have the skills to distinguish forged headers from legitimate headers using today's email clients.

Users can not reliably distinguish legitimate email and website content from illegitimate content that has the same "look and feel". If images and logos are mimicked perfectly, sometimes the only cues that are available to the user are the tone of the language, misspellings or the simple fact that large amounts of personal information is being requested.

Users can not reliably parse domain names. Often they are fooled by the syntax of a domain name through "typejacking" attacks, which substitute letters that may go unnoticed (e.g. www.paypai.com and www.paypal.com), or when numerical IP addresses are used instead of text. The semantics of a domain name can also confuse users. (e.g., users can mistake www.ebaymembers- security.com as belonging to www.ebay.com). Legitimate organizations heighten this confusion by using non-standard naming strategies themselves (e.g., Citibank legitimately uses citi.com, citicard.com and accountonline.com). Phishers have also exploited browser vulnerabilities to spoof domain names, for example by taking advantage of non-printing characters and non-ascii Unicode characters.

Users can not reliably distinguish actual hyperlinks from images of hyperlinks. One common technique used by phishers is to display an image of a legitimate hyperlink. When clicked, the image itself serves as a hyperlink to a different rogue site. Even if the actual hyperlink is displayed in the status bar or a browser or email client, many users do not notice it.

Users can not reliably distinguish browser chrome from web page content. Browser “chrome” refers to the interface constructed by the browser around a web page (e.g., toolbars, windows, address bar, status bar). It is hard for users to distinguish an image of a window in the content of a webpage from an actual browser window. This technique has been used to spoof password dialogue windows, for example. Because the spoofed image looks exactly like a real window, a user can be fooled unless he tries to move or resize the window.

5.2 Overview of the Project

The overview of the project is to present novel approach to overcome the ‘fuzziness’ in the e-banking phishing website assessment and propose an intelligent resilient and effective model for detecting e-banking phishing websites. There is a significant relation between the two phishing website criteria's *(URL & Domain Identity)* and *(Security & Encryption)* for identifying e-banking phishing website. Also we found insignificant trivial influence of the *(Page Style & Content)* criteria along with *(Social Human Factor)* criteria for identifying e-banking phishing websites.

5.3 Module

The project has the following module implementations

1. Extracting Phishing Characteristics Attribute
2. Fuzzification
3. Rule Generation using Associative Classification Algorithms
4. Aggregation of the rule outputs
5. Defuzzification

5.3.1 Modules Description

1. Extracting Phishing Characteristics Attribute

Two publicly available datasets were used to test our implementation: the "phishtank" from the phishtank.com which is considered one of the primary phishing report collators. The PhishTank database records the URL for the suspected website that has been reported, the time of that report, and sometimes further detail such as the screenshots of the website, and is publicly available.

We use a java program to extract the above features, and store these in database for quick reference. Our goal is to gather information about the strategies that are used by attackers and to formulate hypotheses about classifying and categorizing of all different e-banking phishing attacks techniques. The followings are the details of criteria and phising indicators for each criteria.

URL & Domain Identity

1. Using IP address
2. Abnormal request URL
3. Abnormal URL of anchor

4. Abnormal DNS record
5. Abnormal URL

Security & Encryption

1. Using SSL certificate
2. Certificate authority
3. Abnormal cookie
4. Distinguished names certificate

Source Code & Java script

1. Redirect pages
2. Straddling attack
3. Pharming attack
4. On mouse over to hide the link
5. Server Form Handler (SFH)

Page Style & Contents

1. Spelling Errors
2. Copying website
3. Using forms with submit button
4. Using pop-ups windows
5. Disabling right-click

Web Address Bar

1. Long URL address
2. Replacing similar char for URL
3. Adding a prefix or suffix

4. Using the @ symbol to confuse
5. Using hexadecimal char codes

Social Human Factor

1. Emphasis on security
2. Public generic salutation
3. Buying time to access accounts

2. Fuzzification

In this step, linguistic descriptors such as high, low, medium, for example, are assigned to a range of values for each key phishing characteristic indicators. Valid ranges of the inputs are considered and divided into classes, or fuzzy sets. For example, length of URL address can range from 'low' to 'high' with other values in between. We cannot specify clear boundaries between classes. The degree of belongingness of the values of the variables to any selected class is called the degree of membership; Membership function is designed for each phishing characteristic indicator, which is a curve that defines how each point in the input space is mapped to a membership value between [0, 1]. Linguistic values are assigned for each phishing indicator as low, moderate, and high while for e-banking Phishing website risk rate as very legitimate, legitimate, suspicious, phishy, and very phishy (triangular and trapezoidal membership function). For each input their values ranges from 0 to 10 while for output, ranges from 0 to 100. An example of the linguistic descriptors used to represent one of the key phishing characteristic indicators (URL Address Long).

3. Rule generation using associative classification algorithms

To derive a set of class association rules from the training data set, it must satisfy certain user-constraints, i.e support and confidence thresholds. Generally, in association rule mining, any item that passes MinSupp is known as a frequent item. We recorded the prediction accuracy and the number of rules generated by the classification algorithms and a new associative classification MCAR algorithm. Error rate comparative having specified the risk of e-banking phishing website and its key phishing characteristic indicators, the next step is to specify how the e-banking phishing website probability varies. Experts provide fuzzy rules in the form of if...then statements that relate e-banking phishing website probability to various levels of key phishing characteristic indicators based on their knowledge and experience. On that matter and instead of employing an expert system, we utilized data mining classification and association rule approaches in our new e-banking phishing website risk assessment model which automatically finds significant patterns of phishing characteristic or factors in the e-banking phishing website archive data

4. Aggregation of the rule outputs

This is the process of unifying the outputs of all discovered rules. Combining the membership functions of all the rules consequents previously scaled into single fuzzy sets (output).

5. Defuzzification

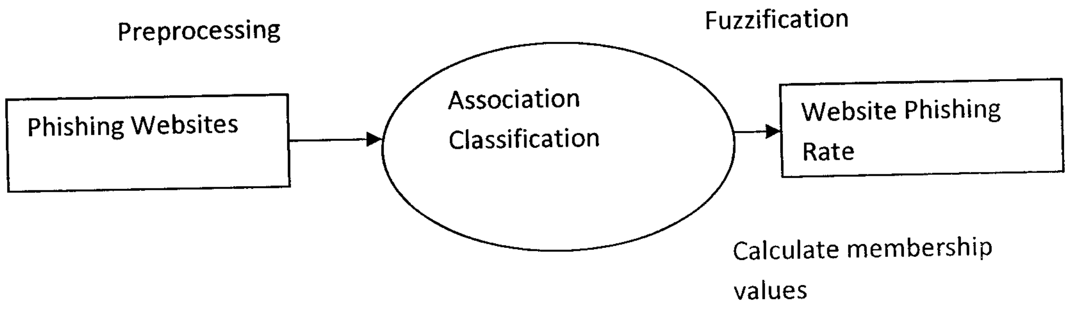
This is the process of transforming a fuzzy output of a fuzzy inference system into a crisp output. Fuzziness helps to evaluate the rules, but the final output has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a number. This step was done using centroid technique since it is a commonly used method. The output is e-banking phishing website risk rate and is defined in fuzzy sets like 'very phishy' to 'very legitimate'. The fuzzy output set is then defuzzified to arrive at a scalar value.

5.4 Pseudocode web phishing

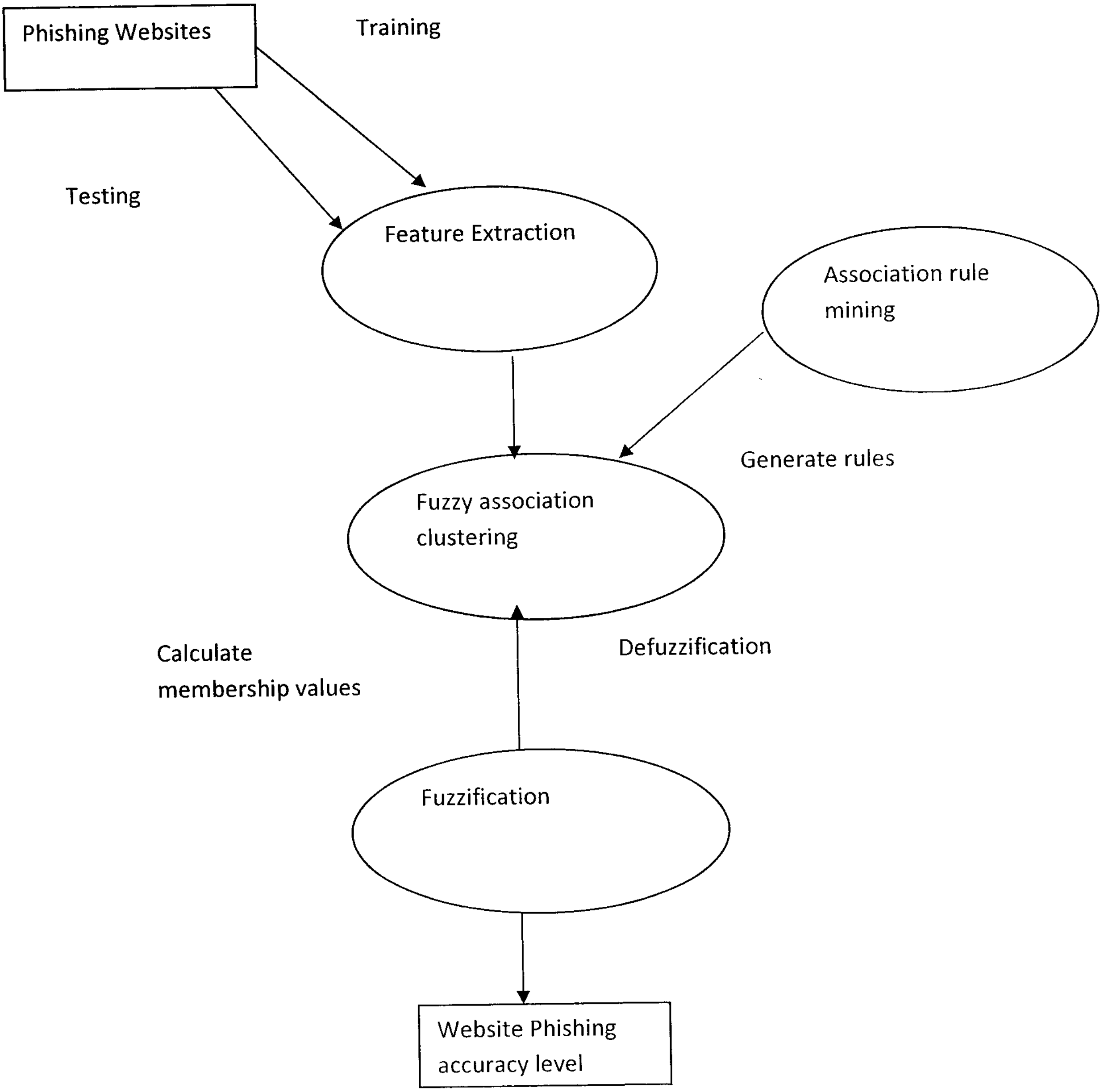
```
Input: Webpage URL
Output: Phishing website identification
Step 1: Read web phishing URL
Step 2: Extract all 27 feature
Step 3: For each feature
            Assign fuzzy membership degree value
            Create fuzzy data set
Step 4: Apply association rule mining & generate classification rule
Step 5: Aggregate all rule above minimum confidence
Step 6: Defuzzification of fuzzy values into original values
Step 7: Apply rule on test data and find whether the site is phishing or not
```

5.4 Data Flow Diagram

Level 0 :



LEVEL 1


Phishing Websites
Training
Testing
Feature Extraction
Association rule mining
Generate rules
Fuzzy association clustering
Calculate membership values
Defuzzification
Fuzzification
Website Phishing accuracy level

5.7 Input Design

Input design is the part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the number of people involved; it is the point of most contact for the users with the computer system; and it is prone to error. If data going into the system are incorrect, then the processing and output will magnify these errors.

The data that are to be inserted are to be inserted with care as this plays a very important role. In order to get the meaningful output and to achieve good accuracy the input should be acceptable and understandable by the user.

There are various approaches for entering data through terminals. This project is implemented as a JAVA application. The essential things that are considered during input design are:

- **The content of the input records:** Data items from the inputs will be used to produce the output. In the project the password that is given as the input is used to authenticate the user.
- **Design of the source document:** Source data is usually recorded in order to standardize the format of input data. Properly designed source documents also aid accuracy and checking procedures.
- **User interface design:** User input is usually through keyboard i.e., the password is entered through keyboard. The submission of the password can be done by either keyboard or by mouse.

- **Volume and frequency of input:** This will dictate the method of input. Usually small volumes of data are input using VDU and thus here too it is input using VDU, validated and processed immediately.

5.8 Output Design

Output design plays a very important role in a system. Getting a correct output is a task that has to be concentrated, as a system is validated as a correct one only if it gives the correct output according to the input.

Computer output is most important and direct source of information to the user. Efficient and intelligent output design should improve the systems relationship with the user and helps in decision making. The output devices to consider depend on factors such as compatibility of the device with the system, response time, requirements and so on.

CHAPTER 6

6. SYSTEM TESTING

6.1 Unit testing

Unit testing is the testing of individual hardware or software units or groups of related unit. Using white box testing techniques, testers (usually the developers creating the code implementation) verify that the code does what it is intended to do at a very low structural level. For example, the tester will write some test code that will call a method with certain parameters and will ensure that the return value of this method is as expected. Looking at the code itself, the tester might notice that there is a branch (an if-then) and might write a second test case to go down the path not executed by the first test case. When available, the tester will examine the low-level design of the code; otherwise, the tester will examine the structure of the code by looking at the code itself. Unit testing is generally done within a class or a component.

6.2 Acceptance testing

After functional and system testing, the product is delivered to a customer and the customer runs black box acceptance tests based on their expectations of the functionality. Acceptance testing is formal testing conducted to determine whether or not a system satisfies its acceptance criteria (the criteria the system must satisfy to be accepted by a customer) and to enable the customer to determine whether or not to accept the system. These tests are often pre-specified by the customer and given to the test team to run before attempting to deliver the product. The customer reserves the right to refuse delivery of the software if the acceptance test cases do not pass. However, customers are not trained software testers. Customers generally do not specify a "complete" set of acceptance test cases. Their test cases are no substitute for creating your own set of functional/system test cases. The customer is probably very good at specifying at most one good test case for each requirement. As you will learn below, many more tests are needed. Whenever possible, we should run customer acceptance test cases ourselves so that we can increase our confidence that they will work at the customer location.

6.3 Test Cases

Unit testing

A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logical. Syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. These errors are shown through error message generated by the computer. For logic errors the programmer must examine the output carefully. When a program is tested, the actual output is compared with the expected output. When there is a discrepancy the sequence of instructions must be traced to determine the problem. The process is facilitated by breaking the program into self-contained portions, each of which can be checked at certain key points .The idea is to compare program values against desk-calculated values to isolate the problems.

Functional testing

Functional testing of an application is used to prove the application delivers correct results, using enough inputs to give an adequate level of confidence that will work correctly for all sets of inputs. The functional testing will need to prove that the application works for each client type and that personalization function work correctly.

Test case no	**Description**	**Expected result**
1	Test for all modules	All module should communicate in the application.
2	Test for every functions in a framework as it display all results.	The result after execution should give the accurate result.

Non-functional testing

This testing used to check that an application will work in the operational environment.

Non-functional testing includes:

- Load testing
- Performance testing
- Usability testing
- Reliability testing

Performance testing

Test case no	Description	Expected result
1	This is required to assure that an application perforce adequately, having the capability to handle many peers, delivering its results in expected time and using an acceptable level of resource and it is an aspect of operational management.	Should handle large input values, and produce accurate result in a expected time

Reliability testing

Test case no	Description	Expected result
1	This is to check that the system is rugged and reliable and can handle the failure of any of the components involved in provide the application.	In case of failure of the system an error handling function should take over the process

CHAPTER 7

7. SYSTEM IMPLEMENTATION

System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. Planning means deciding on the method and the time scale to be adopted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments and involves the following

a. The implication of system environment;

b. Self selection and allocation for implementation tasks;

c. Consultation with unions and resources available;

d. Standby facilities and channels of communication

Training

To achieve the objectives and benefits from computer based system, it is essential for the people who will be involved to be confident of their role in new system.This involves them in understanding overall system and its effect on the organization and in being able to carry out effectively their specified task.So training must take place at an early stage.Training sessions must give user staff, the skills required in their new jobs.The attendance to sort out any queries.

Implementation procedures

The implementation plan consists of the methods for changing from the old system to new one. We will discuss the methods for conversion and procedures used to ensure that it is performed properly. There are several methods for available handling the implementation and consequent conversion from old to the new computerization system. The most secure method for converting from the old to the new system is to run both old and new system in parallel. In this approach personal may operate in the manual order processing system in the accustomed manner as well as start operating the new computerization system. This method offers high security, because even if there is a flaw in the computerization system we can depend on the manual system. However, the cost for maintaining two systems in parallel is very high.

User training

The implementation of the proposed system includes the user training of system operator. Training the system operators includes not only instructions in how to use the equipment, but also in how to diagnose malfunctions and in what steps to take when they occur. So proper training should be provided to the system operations in the proposed system. No training is complete without familiarizing users with simple system maintenance activity. Since the proposed system is developed in a non-GUI. There are different types of training. We can select off-site training to give depth knowledge to the operators.

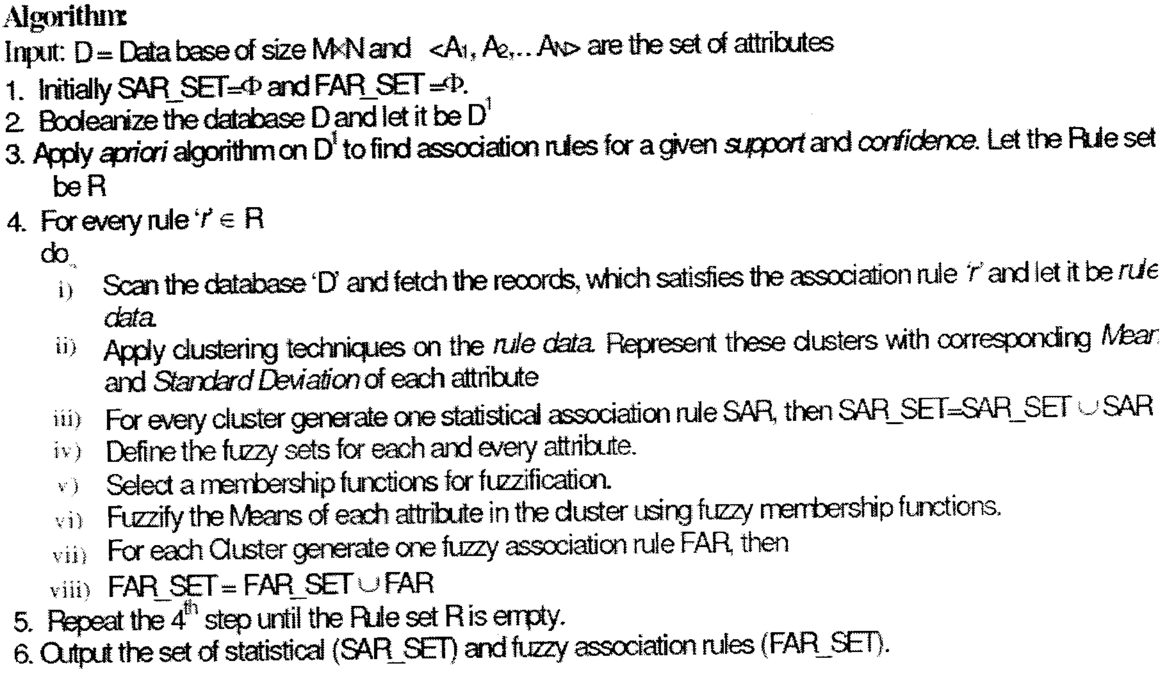
The success of the system depends on the way in which it is operated and used. Therefore the quality of training given for the operating personal affects the successful implementation of the system. The training must ensure that the personal can handle all the possible operations. Training must also include data entry personal. They must also give training for the installation of new hardware. Terminals, how to power there system, how to in down, how to detect the malfunctions how to solve problems etc. the operators must also be provided with the knowledge of trouble shooting which involves the determination of the cause of the problem.

Operational documentation

An operational manual is used as a permanent reference document to inform the computer operations department of the system to be implements, the work to be done in its routine operation, and any special features. The manual is the formal communication of system details to the operations department, but is not the only communication needed. It is essential that provisional details be supplied to the operations department as soon as they are available to give opportunity for preparation of preliminary schedules and forward loading plans and for training and familiarization. The contents should be clear and practical. As it may be necessary for the manual to be partitioned to the requirements, its structure should be determined in consultation with the operations manager. It should be designed to enable problems of operations to be solved without continual reference to programmers or system analysts.

7.1 METHODOLOGY

7.1.1 Fuzzy associative rule mining algorithm



Algorithm:

```
Input: D = Data base of size M×N and  <A1, A2,.. AN> are the set of attributes
1. Initially SAR_SET=Φ and FAR_SET =Φ.
2. Booleanize the database D and let it be D1
3. Apply apriori algorithm on D1 to find association rules for a given support and confidence. Let the Rule set
   be R
4. For every rule 'r' ∈ R
   do,
     i)    Scan the database 'D' and fetch the records, which satisfies the association rule 'r' and let it be rule
           data.
     ii)   Apply clustering techniques on the rule data. Represent these clusters with corresponding Mean
           and Standard Deviation of each attribute
     iii)  For every cluster generate one statistical association rule SAR, then SAR_SET=SAR_SET ∪ SAR
     iv)   Define the fuzzy sets for each and every attribute.
     v)    Select a membership functions for fuzzification.
     vi)   Fuzzify the Means of each attribute in the cluster using fuzzy membership functions.
     vii)  For each Cluster generate one fuzzy association rule FAR, then
     viii) FAR_SET= FAR_SET ∪ FAR
5. Repeat the 4th step until the Rule set R is empty.
6. Output the set of statistical (SAR_SET) and fuzzy association rules (FAR_SET).
```

CHAPTER 8

8. CONCLUSION & FUTURE ENHANCEMENTS

8.1 Conclusion

In this project, we propose the associative classification data mining e-banking phishing website model and show the significance of the phishing websites based on two criteria's (URL & Domain Identity) and (Security & Encryption) with insignificant trivial influence of some other criteria like 'Page Style & content' and 'Social Human Factor' in the final phishing rate, which can help us in building website phishing detection system.

8.2 Future Enhancements

In future, we can extend the approach to some heuristics to enumerate simple combinations of known phishing sites to discover new phishing URLs. To implement an approximate matching algorithm that dissects a URL into multiple components that are matched individually against entries in the blacklist.

CHAPTER 9

9 APPENDIX

9.1 Source Code

```
import java.io.*;
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;
import java.math.BigInteger ;
import java.net.*;
import java.sql.*;
import java.util.*;
public class MainProcess extends JFrame implements ActionListener
{
        BufferedWriter bw;
        JPanel pan=new JPanel();
        public static JTextArea ta=new JTextArea(" ",300,300);
        public static JScrollPane jsp=new JScrollPane(ta);
        JMenuBar bar1=new JMenuBar();
        JMenu startmenu=new JMenu("Start");
JMenu promenu=new JMenu("Process");
JMenu fuzzy=new JMenu("Fuzzy Process");
JMenu lazzy=new JMenu("Lazzy Pruning");
JMenu perform=new JMenu("Performance");
JMenuItem graph=new JMenuItem("Graph");
JMenuItem comp=new JMenuItem("Compare");
JMenuItem startok=new JMenuItem("Ok");
JMenuItem train=new JMenuItem("Training");
JMenuItem test=new JMenuItem("Testing");
JMenuItem train2=new JMenuItem("Training");
JMenuItem test2=new JMenuItem("Testing");
JMenuItem fa=new JMenuItem("Find Accuracy");
JMenuItem fa1=new JMenuItem("Find Accuracy");
static double testacc=0;
static double acc=0;
static double FAccu=0;
double ms=0;
double rr=0;
JButton fileok=new JButton("Ok");
database_conn dc=new database_conn();
fuzzytesting m1=null;
int NOTF=4;
public MainProcess()
```

```
{
}
public MainProcess(int k)
{
          try
          {
                    pan.setLayout(null);
                    add(pan);
                    pan.add(bar1);
                    jsp.setBounds(100,50,800,500);
                    setLocation(10,10);
                    bar1.setBounds(0,0,1000,30);
                    bar1.add(startmenu);
                    bar1.add(promenu);
                    bar1.add(perform);
                    startmenu.add(startok);
                    promenu.add(fuzzy);
                    //promenu.add(aoc);
                    promenu.add(lazzy);
                    promenu.add(comp);
                    fuzzy.add(train);
                    fuzzy.add(test);
                    perform.add(graph);
```

```
        fuzzy.add(fa);
        //lazzy.add(train2);
        lazzy.add(test2);
        lazzy.add(fa1);
        setSize(1000,730);
        setVisible(true);
        promenu.setEnabled(false);
        fuzzy.setEnabled(false);
        train.setEnabled(false);
//      test.setEnabled(false);
        lazzy.setEnabled(false);
        train2.setEnabled(false);
        test2.setEnabled(false);
        fa.setEnabled(false);
        comp.setEnabled(false);
        perform.setEnabled(false);

        train.addActionListener(this);
        train2.addActionListener(this);
        graph.addActionListener(this);

        test.addActionListener(this);
        test2.addActionListener(this);
```

```
                        comp.addActionListener(this);
                        fuzzy.addActionListener(this);
                        fa.addActionListener(this);
                        fa1.addActionListener(this);
                        startok.addActionListener(this);
                        setTitle("Associative Classification Techniques for predicting e-
Banking Phishing Websites");
                   }
                   catch(Exception gg)
                   {
                          gg.printStackTrace();
                   }
              }//end of constructor
              public void actionPerformed(ActionEvent ae)
              {
                   if(ae.getSource()==graph)
                   {
                                        String arr[]=new String[3];
                                        arr[0]="Accuracy";
                                        arr[1]=String.valueOf(acc);
                                        arr[2]=String.valueOf(FAccu);
                                        ChartPanel1.main(arr);
                   }
```

```
                    if(ae.getSource()==comp)
                    {
                            compare();
                            comp.setEnabled(false);
                            perform.setEnabled(true);
                    }
                    if(ae.getSource()==fa1)
                    {
                                                    try
                                                    {
                                                            pan.add(jsp);
                                                            File f=new
File("Files\\lazzyTestingOP.txt");
                                                            if(f.exists())
                                                            {
                                                              f.delete();
                                                            }
                                                            ta.append("Process Of
Testing Using Associative Classification Techniques Is Going On"+"\n");
                                                            lazytesting m1=new
lazytesting();
```

```
ResultSet
rs1=dc.st.executeQuery("select * from sitelist");
int row=0;
while(rs1.next())
{
        row++;
}
System.out.println("Total
Row "+row);
ResultSet
rs=dc.st2.executeQuery("select * from sitelist");
String  arr[]=new
String[row];
int i=0;
while(rs.next())
{
        arr[i]=rs.getString(1);
        i++;
}
for(int j=0;j<NOTF;j++)
{
        m1.startfun(arr[j],2,ms,rr);
System.out.println("Called..."+(j+1));
                               }
                               //test.setEnabled(false);
                               findaccuracy1();
                               fa1.setEnabled(false);
                               comp.setEnabled(true);
                           }
                           catch(Exception kk)
                           {
                           }
        }
        if(ae.getSource()==fa)
        {
            try
            {
                m1=new fuzzytesting();
                pan.add(jsp);
                File f=new File("Files\\TestingOP.txt");
                if(f.exists())
                {
                  f.delete();
                }
```

```
                                        ta.append("Process Of Testing Using Associative
Classification Techniques Is Going On"+"\n");
                                        fuzzytesting m1=new fuzzytesting();


                                        ResultSet rs1=dc.st.executeQuery("select * from sitelist");
                                        int row=0;
                                        while(rs1.next())
                                        {
                                                row++;
                                        }
                                        System.out.println("Total Row "+row);
                                        ResultSet rs=dc.st2.executeQuery("select * from sitelist");
                                        String  arr[]=new String[row];
                                        int i=0;
                                        while(rs.next())
                                        {
                                                arr[i]=rs.getString(1);
                                                i++;
                                        }
                                        for(int j=0;j<NOTF;j++)
                                        {
                                                m1.startfun(arr[j],2);
                                                System.out.println("Called..."+(j+1));
```

```
                  }
                  //test.setEnabled(false);
                  findaccuracy();
              lazzy.setEnabled(true);
              test2.setEnabled(true);
            }
            catch(Exception kk)
            {


            }
        }
        if(ae.getSource()==train2)
        {
              Lazypruning mm=new Lazypruning();
              Lazypruning m1=new Lazypruning(3);
              train2.setEnabled(false);
              test2.setEnabled(true);
        }
        if(ae.getSource()==test2)
        {
                  lazytesting   m1=new lazytesting();
                      String response = JOptionPane.showInputDialog( "Give A
URL For Check" );
```

```
                System.out.println(response);
                response=response.trim();
                String supcount=
JOptionPane.showInputDialog(null,"Enter The Minimum Support Value For Lazy Pruning");
                ms=(Double.parseDouble(supcount));
                String supcount1=
JOptionPane.showInputDialog(null,"Enter The Support Count Value For Lazy Pruning");
                rr=(Double.parseDouble(supcount));
                m1.startfun(response,1,ms,rr);
                test2.setEnabled(false);
                fa1.setEnabled(true);
        }
        if(ae.getSource()==train)
        {
            boolean enter=isInternetReachable();
            if(enter)
            {
                    JOptionPane.showMessageDialog(this,"Now You Are
 Going To Use A Internet Connection To Run This Project");
                    fuzzytraining mm=new fuzzytraining();
                    fuzzytraining m1=new fuzzytraining(3);
                    train.setEnabled(false);
                    test.setEnabled(true);
                    }
                    else
                    {
                            JOptionPane.showMessageDialog(this,"You Must Have A
InterNet Connection To Run This Project");
                    }
                }
                if(ae.getSource()==test)
                {
                    try
                    {
                            m1=new fuzzytesting();
                            String response = JOptionPane.showInputDialog( "Give A
 URL For Check" );
                            System.out.println(response);
                            response=response.trim();
                            m1.startfun(response,1);
                        //test.setEnabled(false);
                        fa.setEnabled(true);

                    }
                    catch(Exception kk)
                    {
                    }
                }
                if(ae.getSource()==startok)
                {
                        promenu.setEnabled(true);
                        fuzzy.setEnabled(true);
                        train.setEnabled(true);
                }
            }
            public static void main(String args[])
            {
                    MainProcess mp=new MainProcess();
                    MainProcess mp1=new MainProcess(4);
            }
            public void compare()
            {
                    ta.append("\n\nAccuracy Produced From Phishing Techniques Is \n\n");
                    ta.append("\t\t"+acc+"%\n\n");
                    ta.append("\n\nAccuracy Produced From lazzy Pruning Techniques Is
\n\n");
                    ta.append("\t\t"+FAccu+"%\n\n");
```

```
        ta.append("****************************************************************
******************************"+"\n");
                    ta.append("\t\tComparing Result...."+"\n");


        ta.append("****************************************************************
******************************"+"\n");
                    ta.append("\n\n\t\tLazzy Pruning Tachniques Is Best\n\n");


        ta.append("****************************************************************
******************************"+"\n");


                    ta.append("\t\t********  Process Finished  *********"+"\n");
              }
              public void findaccuracy1()
                        {
                             try
                             {


                                  System.out.println("In The Process Of Finding
  Accuracy");
                                  Vector v1=new Vector();
```

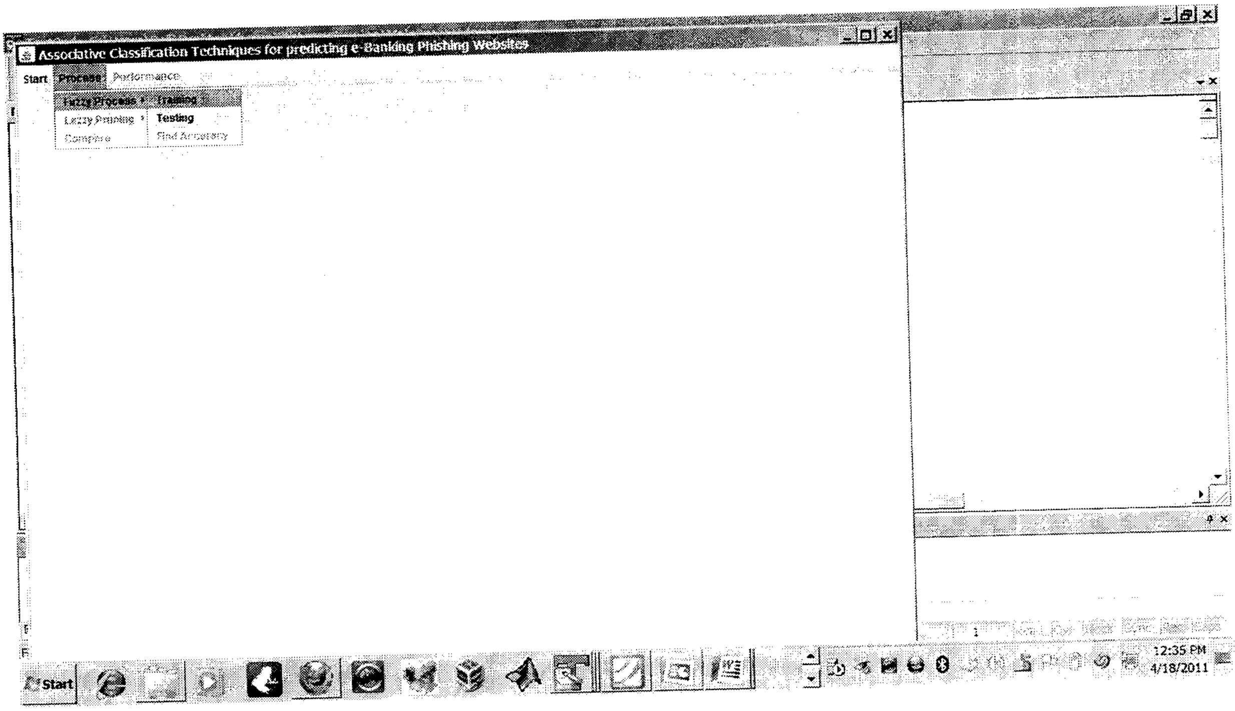
```
                                        Vector v2=new Vector();
                                        BufferedReader test=new BufferedReader(new
FileReader(new File("Files\\lazzyTestingOP.txt")));
                                        String str1=new String();
                                        double count=0;
                                        while((str1=test.readLine())!=null)
                                        {
                                              BufferedReader train=new
BufferedReader(new FileReader(new File("Files\\TrainingOP.txt")));
                                              String str2=new String();
                                              String ass[]=str1.split(" ");
                                              String condi=ass[6];
                                              String val=ass[7];
                                              String webname=ass[8];

                                              while((str2=train.readLine())!=null)
                                              {
                                                    str2=str2.trim();
                                                    String arr[]=str2.split(" ");
                                                    String condi1=arr[6];
                                                    String val1=arr[7];
                                                    String webname1=arr[8];
                                        if(webname.equals(webname1))
                                        {
                                             if(condi.equals(condi1))
                                             {
                                                  count++;
                                             }
                                             else
                                             {
                                             System.out.println("else");
                                             double
res1=Double.parseDouble(val);
                                                  double
res2=Double.parseDouble(val1);
                                                  double diff=0;
                                                  if(res1>res2)
                                                  {
                                                       diff=res1-res2;
                                                  }
                                                  else
                                                  {
                                                       diff=res2-res1;
                                                  }
System.out.println("Differe "+diff);
                                                   if(diff<0.5)
                                                   {

count=count+0.5;
                                                   }
                                                   else
                                                   {
                                                   }
                                              }
                                        }
                                   }//end of inner while loop
                             }//end of outer while loop
                              System.out.println("Count  "+count);
                              System.out.println("Total  "+NOTF);
                              double cc=(double)count;
                              double no=(double)NOTF;
                              double accu=((cc/no)*100);
                         //System.out.println("Acc "+accu);
                              // System.out.println("AccE "+acc);


                               ta.append("\n");
ta.append("*********************************************************************
***********************"+"\n");check(accu);
                                        ta.append("\t\tComparing Result...."+"\n");

ta.append("*********************************************************************
***********************"+"\n");
                                        ta.append("\t\tAccuracy "+FAccu+"\n");

ta.append("*********************************************************************
***********************"+"\n");
                                        System.out.println("Accuracy "+FAccu+" %");
                                  }//end of try
                                  catch(Exception ee)
                                  {
                                  }
                  }
                  public void findaccuracy()
                  {
                        try
                        {

                              System.out.println("In The Process Of Finding Accuracy");
```

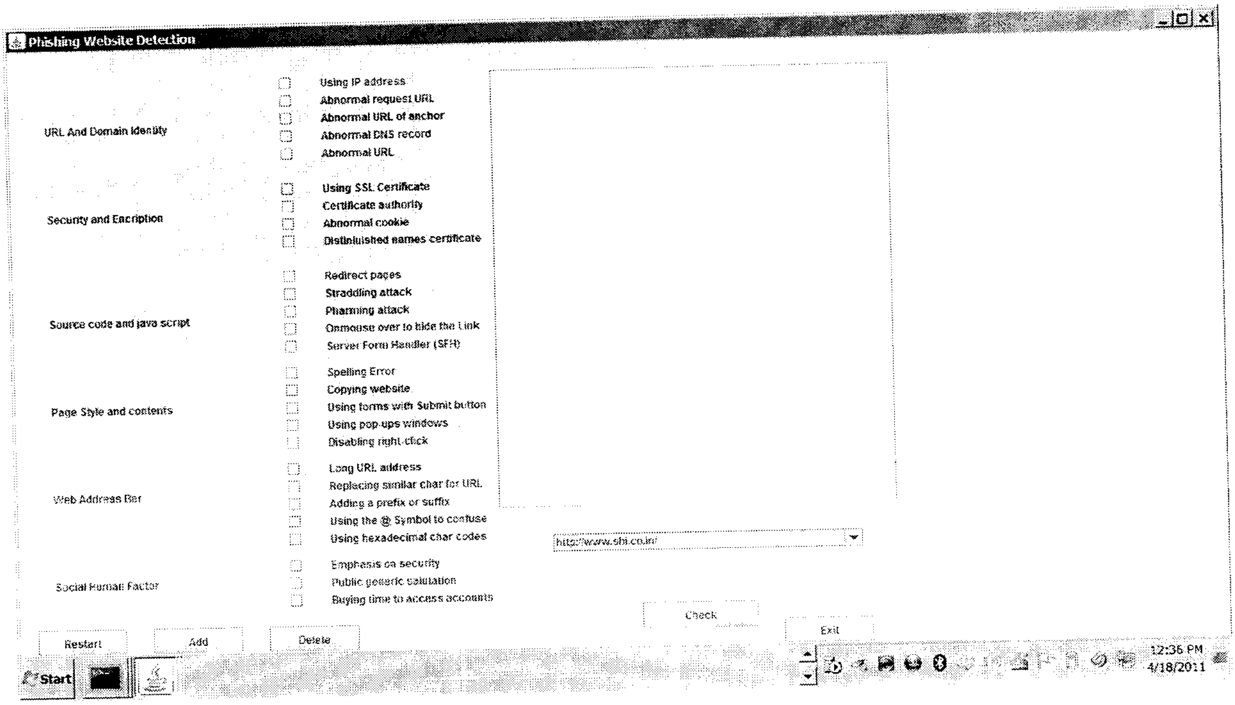
```
                                Vector v1=new Vector();
                                Vector v2=new Vector();
                                BufferedReader test=new BufferedReader(new FileReader(new
File("Files\\TestingOP.txt")));
                                String str1=new String();
                                double count=0;
                                while((str1=test.readLine())!=null)
                                {
                                        BufferedReader train=new BufferedReader(new
FileReader(new File("Files\\TrainingOP.txt")));
                                        String str2=new String();
                                        String ass[]=str1.split(" ");
                                        String condi=ass[6];
                                        String val=ass[7];
                                        String webname=ass[8];


                                        while((str2=train.readLine())!=null)
                                        {
                                                str2=str2.trim();
                                                String arr[]=str2.split(" ");
                                                String condi1=arr[6];
                                                String val1=arr[7];
                                                String webname1=arr[8];
```

```
                        if(webname.equals(webname1))
                        {
                            if(condi.equals(condi1))
                            {
                                count++;
                            }
                            else
                            {
                            System.out.println("else");
                            double res1=Double.parseDouble(val);
                                double
res2=Double.parseDouble(val1);
                                double diff=0;
                                if(res1>res2)
                                {
                                    diff=res1-res2;
                                }
                                else
                                {
                                    diff=res2-res1;
                                }
                                System.out.println("Differe "+diff);
                            if(diff<0.5)
                            {
                                count=count+0.5;
                            }
                            else
                            {
                            }
                        }
                    }
                }//end of inner while loop
            }//end of outer while loop
            System.out.println("Count  "+count);
            System.out.println("Total  "+NOTF);
            double cc=(double)count;
            double no=(double)NOTF;
            double accu=((cc/no)*100);
            ta.append("\n");

ta.append("**********************************************************************
***********************"+"\n");if(accu>90){accu=90.0;}acc=accu;
            ta.append("\t\tComparing Result...."+"\n");
ta.append("***********************************************************************
***********************"+"\n");
                                        ta.append("\t\tAccuracy "+accu+"\n");


ta.append("***********************************************************************
***********************"+"\n");
                                        System.out.println("Accuracy "+accu+" %");
                                }//end of try
                                catch(Exception ee)
                                {
                                }
                        }
                        public void check(double rr)  {
                                 double ds=rr;
                                 double gg=0;
                                 if(ds<=acc){   gg = (double) (Math.random() * (100 - (acc+2)) ) +
   (acc+2);        FAccu=gg;}else{FAccu=ds;}if(FAccu>=100){FAccu=(double) (Math.random() *
   (100 - 98) ) + 98;}      }
                         public static boolean isInternetReachable()
                         {
                                          try
                                          {
```

```
                                        URL url = new URL("http://www.google.com");
                                        HttpURLConnection urlConnect =
(HttpURLConnection)url.openConnection();
      Object objData = urlConnect.getContent();
                            }
                            catch (UnknownHostException e)
                            {
                                  return false;
                            }
                            catch (IOException e)
                            {
                                  return false;
                            }
                            return true;
      }
        }//end of class
```

9.2 Screen Shots





URL And Domain Identity

□ Using IP address
□ Abnormal request URL
□ Abnormal URL of anchor
□ Abnormal DNS record
□ Abnormal URL

Security and Encription

□ Using SSL Certificate
□ Certificate authority
□ Abnormal cookie
□ Distinluished names certificate

Source code and java script

□ Redirect pages
□ Straddling attack
□ Pharming attack
□ Onmouse over to hide the Link
□ Server Form Handler (SFH)

Page Style and contents

□ Spelling Error
□ Copying website
□ Using forms with Submit button
□ Using pop-ups windows
□ Disabling right-click

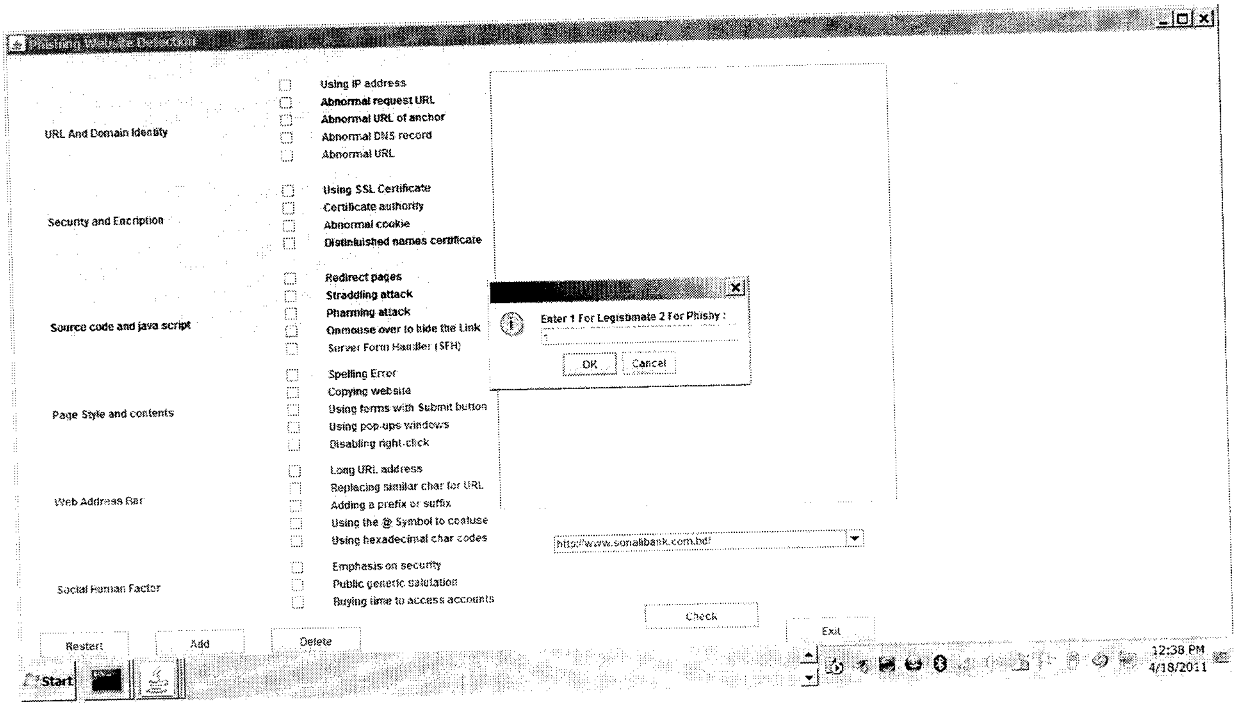
Web Address Bar

□ Long URL address
□ Replacing similar char for URL
□ Adding a prefix or suffix
□ Using the @ Symbol to confuse
□ Using hexadecimal char codes

Social Human Factor

□ Emphasis on security
□ Public generic salutation
□ Buying time to access accounts

http://www.sbi.co.in/



Phishing Website Detection

URL And Domain Identity

□ Using IP address
□ Abnormal request URL
□ Abnormal URL of anchor
□ Abnormal DNS record
□ Abnormal URL

Security and Encription

□ Using SSL Certificate
□ Certificate authority
□ Abnormal cookie
□ Distinluished names certificate

Source code and java script

□ Redirect pages
□ Straddling attack
□ Pharming attack
□ Onmouse over to hide the Link
□ Server Form Handler (SFH)

Page Style and contents

□ Spelling Error
□ Copying website
□ Using forms with Submit button
□ Using pop-ups windows
□ Disabling right-click

Web Address Bar

□ Long URL address
□ Replacing similar char for URL
□ Adding a prefix or suffix
□ Using the @ Symbol to coafuse
□ Using hexadecimal char codes

Social Human Factor

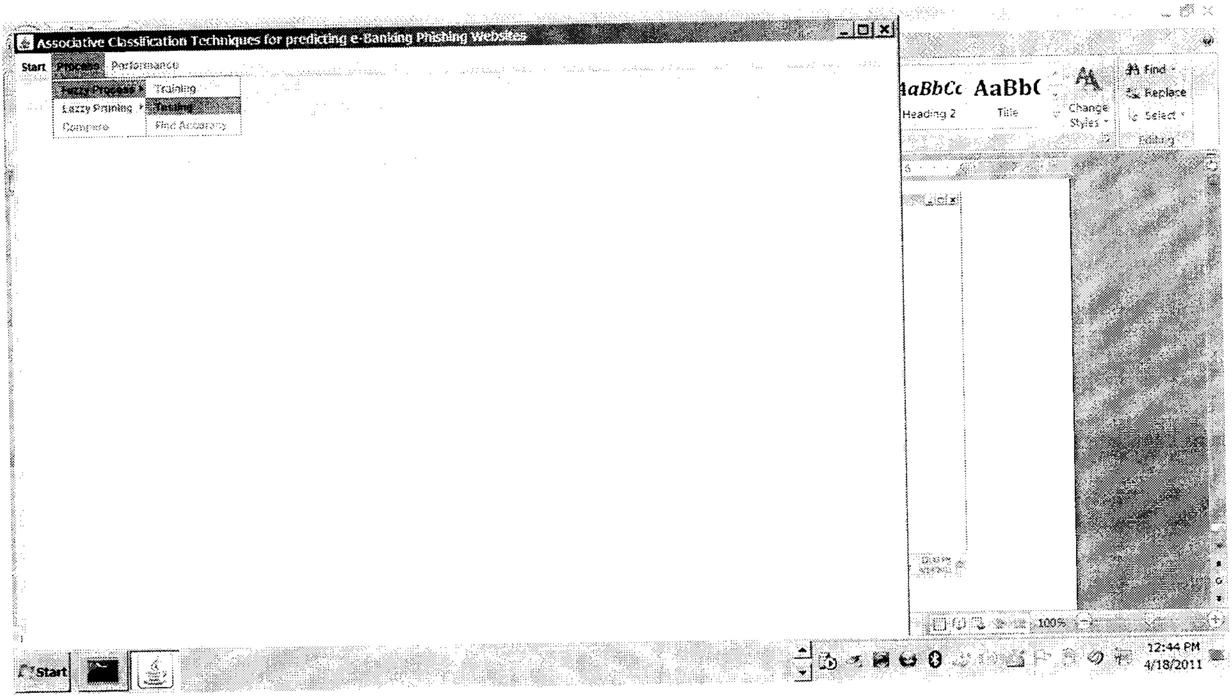
□ Emphasis on security
□ Public generic salutation
□ Buying time to access accounts

Enter 1 For Legistimate 2 For Phishy :

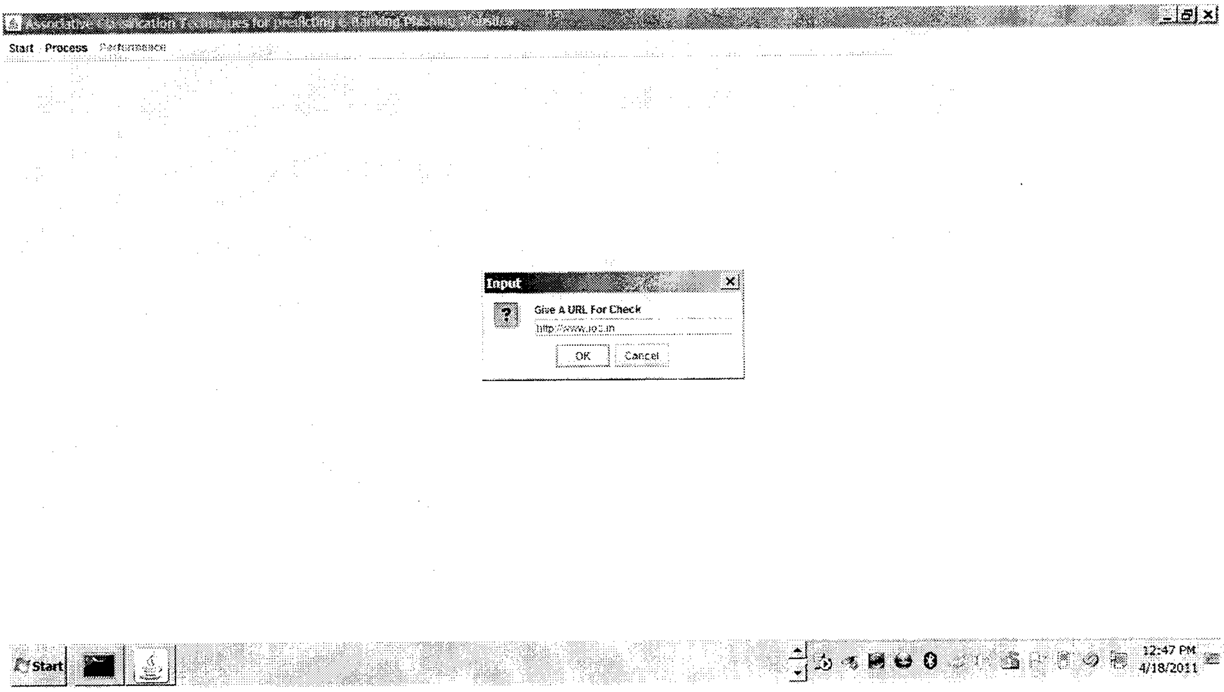
1

http://www.sonalibank.com.bd/

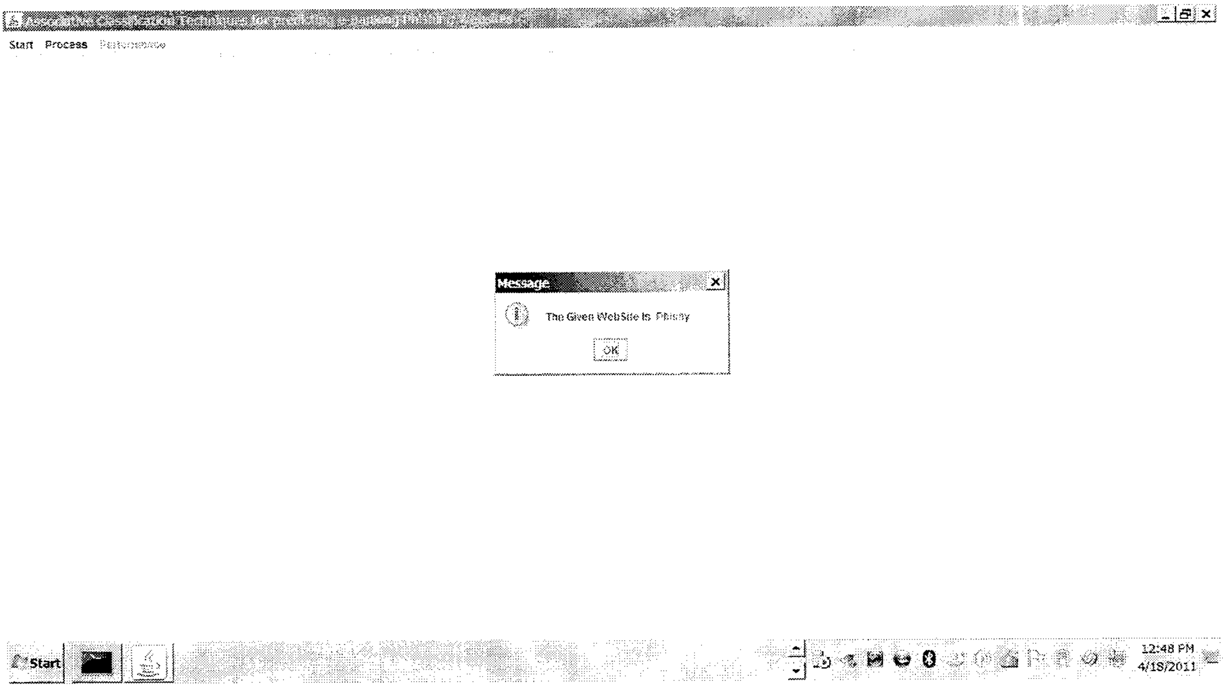
12:38 PM
4/18/2011



Associative Classification Techniques for predicting e-Banking Phishing Websites
Start
Process
Performance
Fuzzy Process
Lazzy Pruning
Compare
Training
Testing
Find Accuracy
AaBbCc
AaBb(
Heading 2
Title
Change Styles
Find
Replace
Select
Editing
100%
Start
12:44 PM
4/18/2011



http://www.ioc.in



The Given WebSite Is Phishy

CHAPTER 10

10 REFERENCES

[1] T. Moore and R. Clayton, "An empirical analysis of the current state of phishing attack and defence", In Proceedings of the Workshop on the Economics of Information Security (WEIS2007)

[2] B. Adida, S. Hohenberger and R. Rivest , "Lightweight Encryption for Email" USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI), 2005,

[3] T. Sharif, "Phishing Filter in IE7," http://blogs,msdn.com/ie/archive/2005 /09/09/463204,as px,,2006,

[4] R. Dhamija and J.D. Tygar, "The Battle against Phishing: Dynamic Security Skins" Proc , Symp. Usable Privacy and Security, 2005.

[5] Microsoft, "microsoft,com/twc/privacv/spam", 2004 .

[6] C. Jackson, D. Simon, D. Tan, and A. Barth, "An evaluation of extended validation and picture-in-picture phishing attacks". In Proceedings of the 2007 Usable Security. www. usablesecurity.orgipapers/jackson.pdf.

[7] Kantardzic and Mehmed. *"Data Mining: Concepts, Models, Methods, and Algorithms,",* John Wiley & Sons.

[8] I.H. Witten and E. Frank, "Data Mining : Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, CA, 2005,

[9] J. Cendrowska., *"PRISM: An algorithm for inducing modular rule",* International Journal of Man-Machine Studies (1987), Vo1.27, No.4, pp.349-370.