Performance Improvement and Analysis of Various Web Browsers using Clustering & Classification

Harish Singh Baghel
Computer Science Department
M.G.C.G.V
Chitrakoot, Satna, India
harishsingh51016@gmail.com

Dr. Bharat Mishra
Physical Science
M.G.C.G.V
Chitrakoot, Satna, India
bm_cgv@redifffmail.com

Abstract— Web Browsers is an intermediate application that can be used to access various files and data over internet. Although there are various Web browsers implemented for the access of various sites over internet, but these browsers contains different features on the basis of which the performance of the web browsers is calculated. Here in this paper analysis of various browsers is computed and compared on the basis of various attributes. The dataset containing attributes is analyzed using FCM clustering and rules are generated using decision tree to compute the efficient web browser.

I. Introduction

The web browser can be considered as user's window to the world which provides an interface to perform a wide range of activity like email, shopping, social networking, finance management, business etc. Similar to client server model a web browser is a web client which runs on users/clients computer requesting web server to operate requests generated by client for obtaining information or resources [1]. The web server in return locates and sends the information to the web browser thereby displaying the results. But browsing web can be a dangerous because all major Web browsers have significant security flaws due to which hackers can attack you when visiting a Web page that contains malicious content. But this also difficult because attackers cannot choose the time and place of the attack, and wait for a victim to come to their Web page making it difficult for attackers to target specific users unless they have information about the user's browsing habits. The browsers have different vulnerabilities so attackers have to choose the attack which fits the browser [2]. Browser is able to analyze and deploy contextual meaning to sensitive information that is provided by the user during his/her personal activities and use.

Browsing the Web opens system to variety of serious attacks. The combination of browser and operating system is vulnerable. There are a number of steps to decrease this vulnerability all of which require some time and effort and some of which limit the usefulness of the Web in various ways. Sites should personally evaluate risks based on their own local criteria and decide the steps that are appropriate for the systems. Browser even has access to the information provided by user while performing browsing activities even when user encrypts all incoming and outgoing communication.

This level of access to sensitive and personal data is high and needs efforts to ensure its complete confidentiality and integrity. Web browser access World Wide Web (WWW) which is a shared information system that operates on top of Internet. Web browsers retrieve content and display the content from remote web servers using a protocol called Hyper-Text Transfer Protocol (HTTP) which as stateless and anonymous. Web browsers access web pages that are written using language called Hyper-Text Markup Language (HTML). The pages are augmented with other technologies like Cascading Style Sheets (CSS) adding layout and style information and JavaScript that allows client-side computation which is executed by web browser. Browsers also provide

other useful features like bookmarking, password management, accessibility features, history etc. [3].
Browsers are capable of creating logs that contain cache, history, cookie and downloads list [4]. These log files also contain private information about the user accessing the web pages and content on internet. With the help some tools these log files can be analyzed and accessed which sometimes is useful and sometimes are security vulnerability or threat.
The browsers are capable of lot of functionalities. Nowadays web applications are being incorporated with web browsers. The applications are implemented in such context that the browsers are able to execute the applications context [5].
The web browsers and application programs access information on World Wide Web. But the primary function of all the web browsers is the same they just differ from each other in some aspect like:

- For Platform: Linux, Windows, Mac, BSD and Unix
- In Protocols: FTP, SAMBA, HTTP, IMAP, etc.
- Through Layout Engine: Amaya, Gecko, Trident, KHTML, WebKit
- Graphical User Interface (GUI), Mobile Compatibility, HTML5 Support, Open Source.

Whereas the general, primary and secondary features of web browsers are downloads, bookmarks, management of passwords etc. and provides functions for spell check, toolbars of search engines for easy access, browsing through tabs, filtering of ads, pop-up blocking etc [6]. Therefore browser requires high accuracy while it's designing and implementation because a small mistake makes it vulnerable to threats and for this understanding of browser vulnerability it is required architectural design knowledge of browsers [7].
The browser consists of three parts the controller, the client program and the interpreter. The controller accepts inputs from the input devices and to access a document uses a client program like http. Ftp, telnet etc. On access of document the controller with the help of interpreter like html, cgi or java etc. display it on the screen. For such functioning of browser vulnerability can be a weakness or design flaw in software program that a attacker uses to decrease performance of system or to get an unauthorized access to private data.

II. LITERATURE REVIEW

M. T. Louw et.al. [1] they examined the security issues in functionality extension mechanisms that are supported by web browsers. Plug-ins in modern web browsers have unlimited access without any restraint and thus are vulnerable for malware. Taking advantage of lack of security mechanisms for browser extensions and they gave a piece of malware for Firefox web browser named as BROWSERSPY which requires no special privileges to be installed. They showed how BROWSERSPY takes complete control of a user's browser space and observes all the activity performed through browser and cab even remain un-detectable. They then defended such malware various defense strategies. They proposed mechanism which through code integrity checking techniques controls the extension installation and loading process and runtime monitors the extension behavior which provides a foundation for defending threats due to installed extensions. They addressed malicious extensions threat by a mechanism through which at load time the installation integrity of extensions is validated and infrastructure for runtime monitoring and policies for prevention of further attacks on browser core integrity and sensitive data confidentiality. With the help of their mechanism the browser allowed only extensions installed by the user to be loaded and detects unauthorized changes made to installed extensions and enabling the browser to analyze and monitor extension code at runtime [1].
A. Guha et.al. [8] they remarked that third party extensions like popup blocking, form filling etc enrich browsers in multiple ways but if these extensions are used improperly then security risks arises as they uses local storage and cross domain network access because many extensions are over privileged under existing architecture of browsers. They proposed a framework for authorization, analyzing, verification and deployment of secure browser extensions. They proposed Data log for specification of fine grained access control and data flow policies that limit the ways of extension use thus restricting its privilege over security. They provided visualization tools for assisting policy analysis, and compilers for translate of source code of extension to either .NET byte-code or JavaScript. They developed a methodology that enforces safety statically by finer grained access control model for browser extensions that characterizes security property for extensions [8].
D. Jang et.al. [9] they gave that Web browsers are capable of accessing valuable private data and act as a mediator for accessing it in various domains are also capable to protect it but attackers continuously exploit browser vulnerabilities to ex-filtrate this private data and take over the control of the system. They gave QUARK browser which has a verified and implemented Coq kernel. Their implementation is in accordance with the specification implies property like cookie integrity, confidentiality, tab non-interference etc. With the help of mechanical proof assistant they achieved security guarantees through formal shim verification. They QUARK browser implemented by them can experiment with additional web policies without the requirement of re-engineering the browser or formalizing its behavior details [9].
M. Silic et.al. [10] gave the idea that web earlier was used just for browsing and reading content but today has become the most used platform for application development. The user's can nowadays crate their own content and have applications built which have impact on user's life which thereby has also increased security concerns. Web browsers are used by everyone to access content on internet. Each web application can be executed inside the web browser. The browser mediates between users and applications due to which malicious applications can easily be loaded and executed inside a web browser and thus making it vulnerable in preserving security of it. Browser's architecture does not provide full protection threats that are related to cross-site attacking, session hijacking and user interface compromise do

not modify. The modern browser sometimes sacrifices compatibility with web to provide high security. For such provisions they described the concept of web program isolation in the browser for new security challenges and performance [10].

A. Barth et.al. [11] remarked that Cross-site scripting defenses HTML documents and does not focus on attacks that involves browser's content sniffing algorithm who treats non-HTML documents as HTML. Involving in these attacks authors can upload malicious papers that can automatically submit stellar self reviews. They developed high fidelity models of the content-sniffing algorithms for systematic study of such attacks. They proposed and implemented a content-sniffing algorithm which provides security and is compatible. To develop high fidelity models of the content-sniffing algorithms they used source code inspection and string-enhanced white-box exploration. They provided filter for web sites based on their model which blocks content sniffing XSS attacks and for other websites they used browser content-sniffing algorithms which avoid privilege escalation and uses prefix-disjoint signatures [11].

C. Grier et.al. [12] for more secure web browsing they proposed and gave a new browser called as OP web browser which improves the state of the art in browser security. They combined operating system design principles with formal methods for a more secure web browser. The design philosophy adopted by them is partitioning the browser into smaller subsystems and making all communication between subsystems explicit and simple. They designed a small browser kernel which manages browser subsystems and interposes on all communications between them. They proposed three security features given as giving flexible security policies allowing to include plug-ins within the security framework. Address bar within browser user interface always show correct address for current web page and browser level information-flow tracking system given enables post-mortem analysis of browser-based attacks. The OP web browser given by them is responsive to user interaction and implements multiple features making it compatible with current web pages [12].

III. PROPOSED METHODOLOGY

1. Take an input dataset contains attributes of browsers.
2. Apply Fuzzy C-means Clustering for the clustering of values of the dataset.
3. Apply CART algorithm to generate rules base don decision tree.
4. Compute the performance of the web browser based on the generated rules.

Fuzzy C-Means Clustering

The Fuzzy C-means clustering uses the concept of categorizing the data in two or more clusters that belongs to the same group as generated in Fuzzy Logics. The objective function used in the Fuzzy Clustering provides the minimization of the function so that the clustering can be done efficiently. The Objective function can be given by:

$$O_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\|x_i - c_j\|2$$

Where m lies between

u_{ij}	It is defined as the degree of membership of , in the present cluster j.
x_i	It is denoted as the ith of the d-dimensional data to be measures in the dataset.
c_j	It is defined as the d- dimension cluster center present in the dataset.

Table Annotations used in the Algorithm

The Algorithm consist of the following few steps along with the minimization of the objective function.

[1] First of all the Dataset whose clustering is done is initialized with the objective function as it is a matrix which contains the values to be clustered of m rows and n columns.

[2] After each p-step compute the centroid of the matrix or vector matrix of the dataset denoted as C,

$$C^{(k)} = [c_j]$$

,and contains the vector Matrix S,

[3] Compute objective function and minimize the effect of the objective function using,

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . xi}{\sum_{i=1}^{N} u_{ij}^{m}}$$

[4] Update each value of S[k] with next S [k+1]

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$$

[5] To check the computer value of S[k] and S[k+1], means ,

$$\|S^{k+1} - S^{k}\| < \epsilon$$

Stop the process otherwise go to Step -2.

CART Algorithm

Step 1: Start with root node (t = 1)

Step 2: Search for a split s* among the set if all possible candidates s that gives the purest decrease in impurity.

Step 3: Split node 1 (t = 1) into two nodes (t = 2, t = 3) using the split s*.

Step 4: Repeat the split search process (t = 2, t = 3) as indicated in steps 1-3 until the tree growing the tree growing rules are met.

IV. RESULT ANALYSIS

Algorithm	Correctly Classified Instances	Incorrectly Classified Instances	Time taken to build model
Naïve Bayes	50%	50%	0.03 sec
J48	72.2222 %	27.7778 %	0.04sec
Random Forest	61.1111 %	38.8889 %	0.04sec
CART	50 %	50%	0.02sec

Table Comparison of various Classification Algorithms

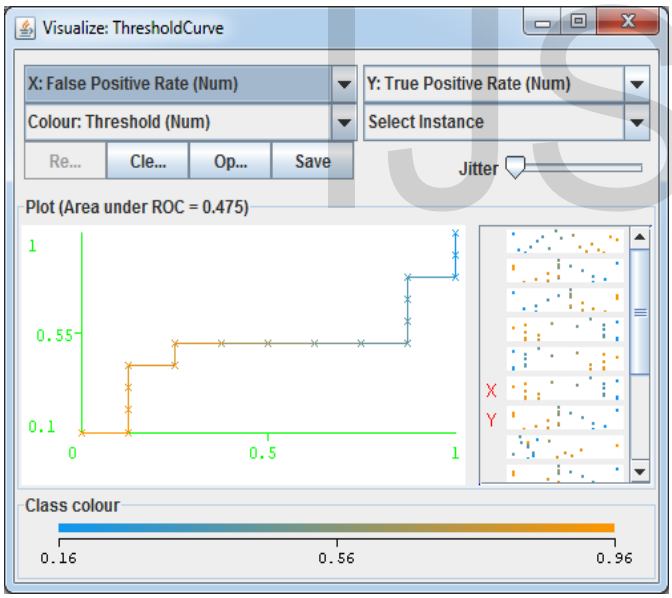

Figure Threshold Curve of Web browsers

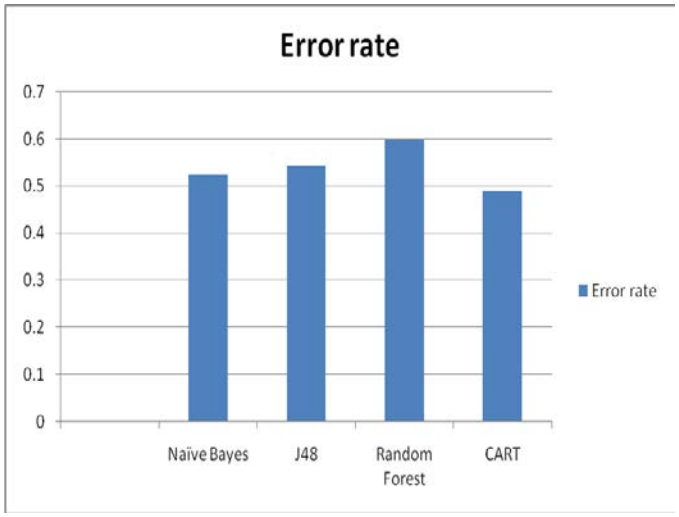

Figure Error Rate of Algorithms

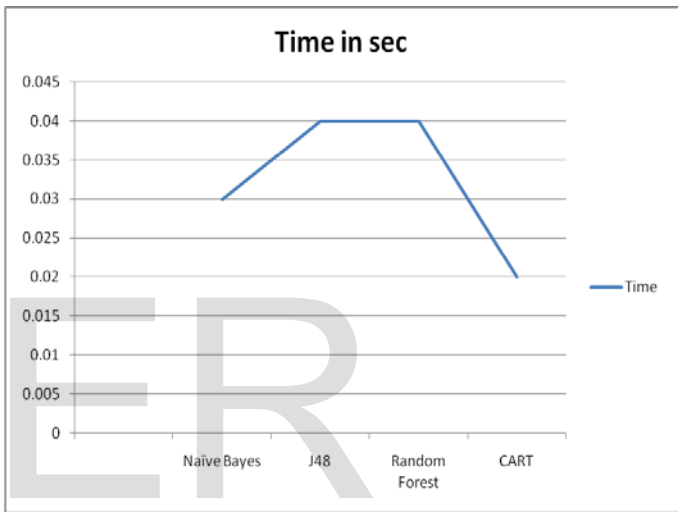

Figure Time Complexity of Algoithms

V. CONCLUSION

The proposed technique implemented here for the analysis and improvement of web browsers using hybrid combination of clustering and classification is given. The technique efficiently enhances the performance of the web browsers by analyzing various features of web browsers. The technique also improves Correctly classified instances of the dataset.

REFERENCES

[1] Mike Ter Louw, Jin Soon Lim, and V.N. Venkatakrishnan "ExtensibleWeb Browser Security", 2007

[2] Michael V. Hayden "Advisory Memorandum on Web Browser Security Vulnerabilities", NSTISSAM INFOSEC 3-00, August 2000

[3] Alan Grosskurth and Michael W. Godfrey "A Reference Architecture for Web Browsers", 2003.

[4] Junghoon Oh, Namheun Son, Sangjin Lee and Kyungho Lee "A Study for Classification of Web Browser Log and Timeline Visualization" Springer-Verlag Berlin Heidelberg, 2012.

[5] Ivan Zuzak , Marko Ivankovic and Ivan Budiselic "A Classification Framework for Web Browser Cross-Context Communication" School of Electrical Engineering and Computing, University of Zagreb, Croatia, 2011.
[6]http://www.buzzle.com/articles/types-of-web browsers.html.
[7] Dhruwajita Devi, Dhrubajyoti Pathak and Sukumar Nandi "Vulnerabilities in Web Browsers", 2010
[8] Arjun Guha, Matthew Fredrikson, Benjamin Livshits and Nikhil Swamy "Verified Security for Browser Extensions", IEEE Symposium on Security and Privacy, 2011
[9] Dongseok Jang, Zachary Tatlock and Sorin Lerner "Establishing Browser Security Guarantees through Formal Shim Verification", 2011
[10] Marin Silic "Security Vulnerabilities in Modern Web Browser Architecture", 2010
[11] Adam Barth, Juan Caballero and Dawn Song "Secure Content Sniffing for Web Browsers or How to Stop Papers from Reviewing Themselves", 2009
[12] Chris Grier, Shuo Tang, and Samuel T. King "Secure web browsing with the OP web browser", 2007.