Volume 3, Issue 6, June 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Phishing Detection System Using Machine Learning and Hadoop-MapReduce

Kaustubh A. Hiwarekar, Dr. R. C. Thool

Department of IT

SGGSIE&T, Nanded

India

Abstract— Detecting & Identifying phishy websites is a tedious work. Several attributes are needed to be taken into consideration & finally using the data mining algorithms, a final decision is made. For finding these attribute we have used MapReduce algorithm in collaboration with Hadoop file system. In existing Online Phishing Detection systems, usually the reference to the database is taken for making any conclusion about the degree of phishiness of the website. In this proposed system, we concentrate on getting the necessary attributes in real time environment using Hadoop-MapReduce, thus increasing both speed & efficiency of the system. This system is very trustful, which surely guarantees that we will not miss a phishy website, even if it is a new-born.

Keywords— Anti-Phishing, Phishing, MapReduce, Hadoop, Machine learning

I. INTRODUCTION

Phishing is a fraudulent attempt, usually made through email, to steal your personal information. Phishing websites are forged websites created by malicious people to mimic real websites. The victims of these phishing websites or emails might expose their personal information like the passwords of their credit card, bank account etc. This normally results into the financial loss of the victim. Currently the systems available in the market like anti-phishing toolbars [1], anti-phishing filters [2] embedded in the antivirus programs, etc. rely mainly on the database obtained by checking the suspicious websites & storing the phishy websites in the database. The procedure involves a number of steps. Firstly the anti-phishing program tries to detect the phishing website on its accord, if it failed to decide about the authenticity of the program, it simply sends the URL to the server for checking the authenticity. The testers check whether the site is original or not, if not, then an entry is made in the database & the updated database is sent to all the known users of the anti-phishing product. The whole process is lengthy, so relying totally on it might prove dangerous. The solution is to check the website attributes, as many possible, in the real time environment. The less the suspicious values of the attributes, more the authenticity of the website, and vice versa. Since the process is done in the real time, the user doesn't need to wait till the update is made & downloaded in proposed system. User is safe from the phishing websites if user uses our proposed system.

Machine learning has found its significant applications in various security related fields like Intrusion detection system, email filtering, social network analysis, image analysis. Likewise in anti-phishing also machine learning is very useful. It is needed to speed up in finding attribute values which can be given by means of Hadoop-MapReduce. MapReduce using Hadoop is also having significant application in many ways. It is used mostly in cloud services EC2. Hadoop-MapReduce is mainly used to handle large sized files. Here we are going to use this quality of Hadoop-MapReduce in making fast decision about the incoming website URLs are phishy or not.

II. RELATED WORK

Phishing is direct attack on identity of user, attacker steals identity of user and impersonate as that victim user. So it is way too different than the virus, malware attacks. It is more of user specific attack so security need to be provided at user level. For user level security toolbars [1] are developed as add-on to the browsers. Netcraft [1] toolbar for the Mozilla browser. Mostly work of these toolbars is to just send URL to their respective servers where all necessary processing is done. After finding the result it is sent back to the toolbar which indeed displays the result in that respective browser. This process takes considerable amount of time to reduce this real time processing at end user is necessary. The other way is to use of Black Lists by the browsers. Black List is maintained by browsers like Google Crome (Google Safe Browsing) [3]. These Black Lists are updated to time manually by hiring expert who manually categories suspected URL whether they are Genuine of phishing. These updates may take little time as it is done manually and updated manually. Other browsers also use same technic for anti-phishing. Anti-phishing working Group (AWPG) helps its partners to build anti-phishing solutions. AWPG generates monthly reports [4] about the current phishing activities. It keeps an eye on phishing activities all over the world and collaborates with its partners [5] with the information obtained. Another main contributor to the anti-phishing work is phishtank.com [6]. It provide free corpus of the current active phishing websites as well history. This corpus also contains detailed information about that phishing website. Phishtank.com corpus is

updated by the volunteering users who report phishing websites. This corpus is available for free for developers developing applications. There are other techniques for identification of phishing web-pages include image processing [7]. In this technic snap-shot image of the webpage is compared with the original web-pages of legitimate websites. The original sites show up with certain logo characteristics which helps them to differentiate from the duplicate phishing websites. The webpage image is divided into block and block by block characters are cross checked with the certain base line and reach to the conclusion. In case of Logo based watermarks in those logos are checked. Server based techniques directly scan E-mail servers, domain servers which host websites, DNS servers which resolve the URLs [8]. E-mail server based technique extract all the suspected URLs from inbox as well as from spams and examine them as most attacked users are from phishing E-mails.

III. SYSTEM DESIGN

Framework below in figure (1) shows proposed framework. The user will enter the URL of the webpage, she wishes to visit. Using that URL, we will download the source code of the webpage & then decide the values of the attributes. For finding these values we will make use of Hadoop-MapReduce [9]. This will speed up the process of attribute value assignment. Basic word count example [10] of Hadoop-MapReduce is used to search sensitive words in webpages. In same way wherever required help of Hadoop is taken. These calculated attributes are the input to the Prediction module. Based on the records stored from phishtank.com database, training data is prepared. All the characteristics of reported phishing website at phishtank.com corpus are studied and based on that attributes are decided and training data for machine learning algorithm is prepared. Using training data machine learning algorithm generates set of rules based on which decision is to be made. Prediction module gets two inputs rules generated by machine learning algorithm and attribute found from requested URL. Prediction module finally predict URL falls under which category (Phishing, Legitimate, and Doubtful).

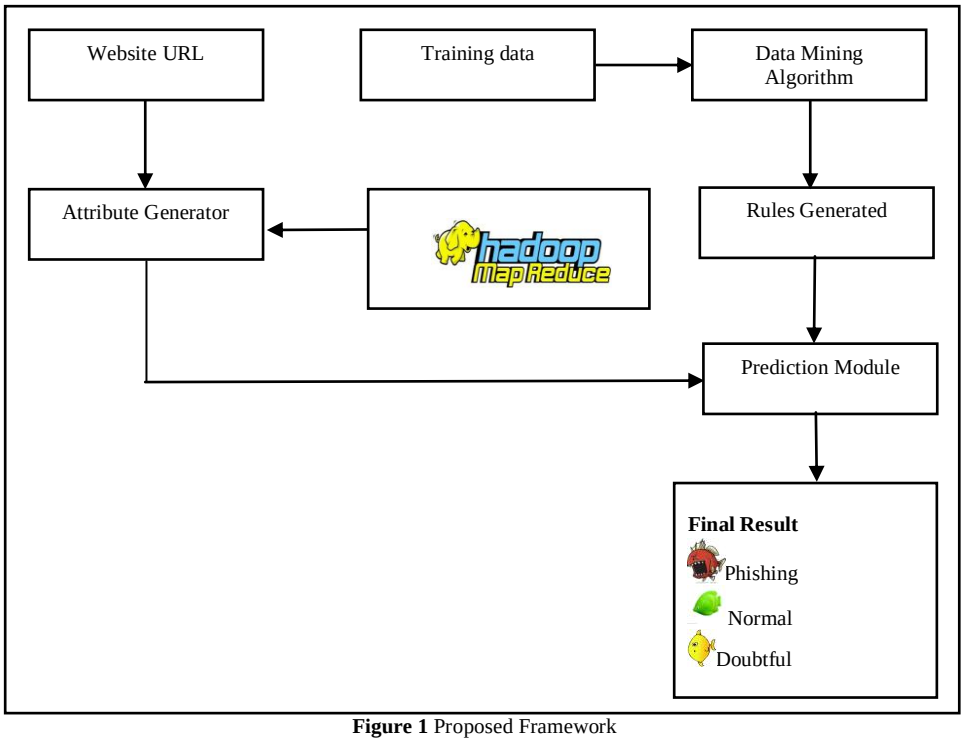


Figure 1 Proposed Framework

IV. ATTRIBUTES CONSIDERATION

While concentrating on getting the required attributes, which will be enough to decide the phishiness of the website, we searched lot of documents, formulated a few on our own. But since the attackers are quite advanced these days, we needed to consider the visual aspects too, along with the usual coding methods. We got this architectural model in (shown in figure (2)) the Intelligent Phishing Detection System for e-banking Using Fuzzy Data Mining [11]. As it fulfilled all the requirements, we didn't modify it, instead concentrated on making it efficient and speedy. The three layers act as the backbone of the system. The layer manager part of prediction module acts as the brain of the system, making the decisions there itself. As the system is not limited to specific use, you can use it for any general purpose. The twenty seven attributes as seen in the figure, are calculated in real time & these values might be different from the values calculated for the same website, even if slight changes are made to the website. Thus the user can remain free from a suspicious website, although it was previously listed as the authentic one. In order to simplify the system, we have simply kept all the three layers at the same priorities, thus ensuring that even the less important factors can take part in decision making. This helps in detecting the suspicious websites. There are some attributes which need only word count whether that word is present in source code or not. For example ATM PIN is restricted word then we just need to find out in source code of page word is present or not and assign value to attribute. These situations are very often in anti-phishing systems based on machine learning.

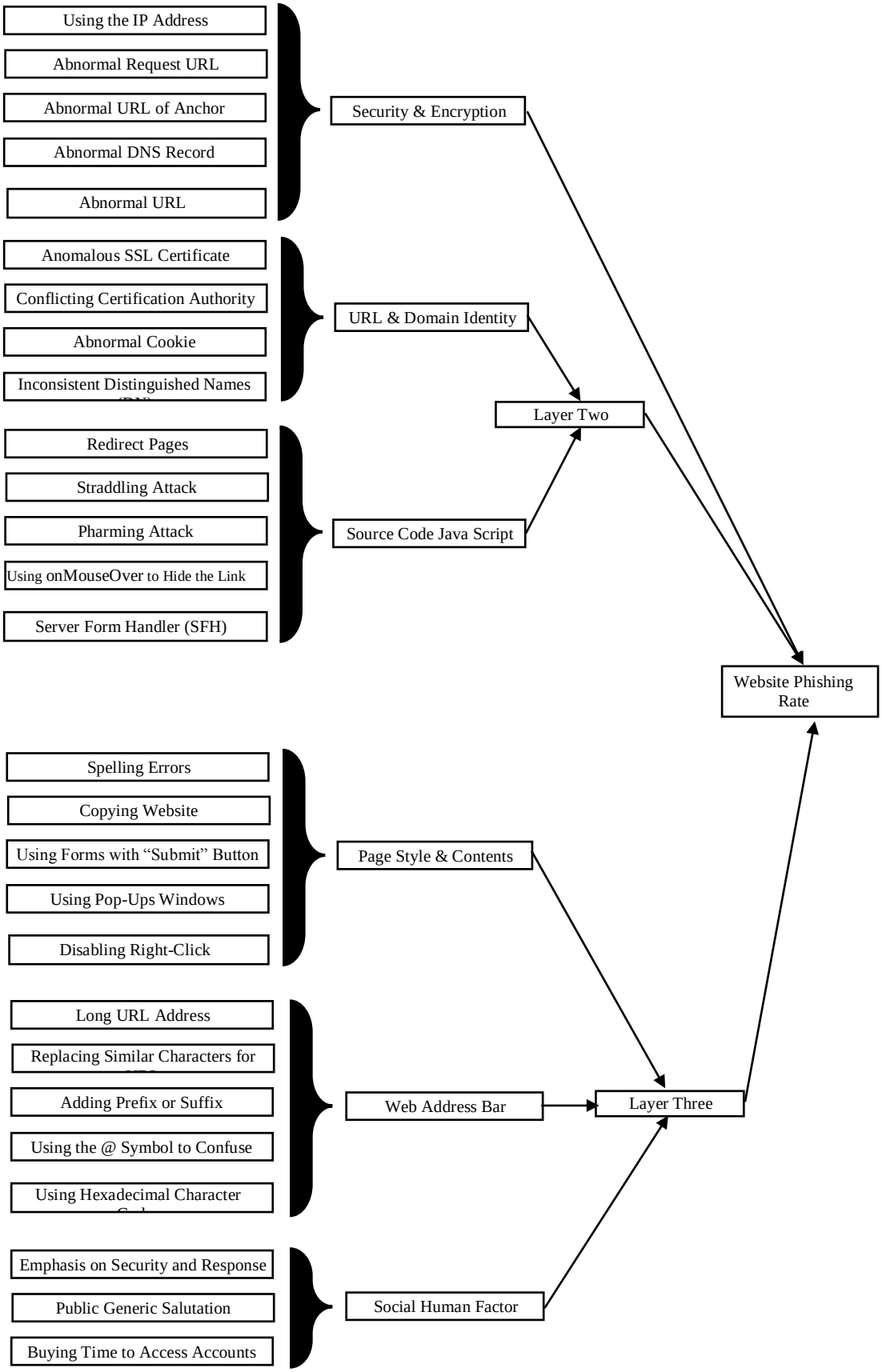


Figure 2: Detailed architecture of website Attributes Considered

Use of Hadoop is generally made in large datasets. To satisfy this we are considering the scenario of the proxy network. If we consider whole network working under one proxy server. In average some hundred or thousand number of URL are being requested per second at the proxy server. To check Phishiness of all the URL we will need fast internet as well as fast attribute calculator which indeed going to decide the type of website. For the sack of fast attribute calculator we are using Hadoop-MapReduce. All the source code of requested URL will get downloaded in the system and for

finding the attribute value we need whether those tags of words are present in the source code or not. Hadoop will take Source code and the words to be found as input and will produce output within fraction of second the frequency count. Ultimately this will decide the attribute value which will affect the decision accordingly. All the requested URL can be processed simultaneously and produce results at faster speed than before. Accelerating the process of finding attributes is ultimately increase the speed of decision of data mining algorithm.

V. CREATION OF RULES NEEDED

Generating rules is quite tedious job. In order to make this job more reliable we made the use of WEKA [12] data mining tool. With all the data mining algorithms implemented, it proved very helpful in determining the most efficient ones. When actual experiments were done on the database, it was found that although for smaller data sets J4.8 & PRISM prove more efficient but when a dataset comprising of more than 500 entries was taken, PART leaped much ahead of them. So we preferred PART over all others. PART stands for Projective Adaptive Resonance Theory. This proves largely fruitful when it comes to handling larger databases. Even when we used it via WEKA, it gave us efficiency more than 85%, thus building the confidence & increased the reliability of the system. We simply gave the attributes received to the Prediction Module. The layer manager acted as the coordinator between the rules generated & the attributes received. By properly classifying the attribute values & by correctly & hierarchically maintaining the results, the layer manager predicted about the nature of the website, the user intends to visit. For simplicity of guidelines, we classified all the websites into three categories like Phishy, Doubtful & Legitimate. If the website is legitimate, the user can easily visit the website. If it is phishy, she is blocked from visiting the website, thus ensuring that our user will not be a victim of the bait, kept by a malicious user. If the website is suspicious, the decision is to be taken by the user whether to visit the website or return back to earlier page. If she intends to visit the website, she is warned not to disclose any personal information.

VI. DATA SETS AND EXPERIMENTAL SETUP

A. Data Sets

By using previous studies in the intelligent phishing detection systems [11] .arff files for the layers are generated manually. These files are the core for rule generation algorithms which indeed are the data sets of our system there are preferences assigned to the some attributes based on their importance in scope of phishing site building. This can be explained as for example the phishing site for sure contain the submit button so in case of forming the file for related layer we have taken care of the preferences or in other words priorities of the attributes. More the priority more will be the effect of that attribute in the rule generation process. We have used the datasets as shown in the figure (3). These datasets are given as input to the rule generator in implemented in WEKA machine learning data mining tool [13]. The datasets are in hierarchical manner. First of all the sub-layers attributes are found and hierarchically layer wise the final result is generated in last step. For generating the intermediate datasets while finding phishiness of any site we need to simply put the "?" for the last attribute in every layer attribute so that weka tool rule generator will add the predicted value at the place of "?" in dataset given to it, which will be used for the further process in hierarchically next level and so on. At last we get the final result as phishy, Legitimate or Doubtful. Efficiency goes on increasing as the correctly classified instances percentage increases. For that accurate priority based dataset is provided to the rule generator. As the numbers of records in the data set are increased the correctly classified instances are increased. Dataset shown in figure (3) is for layer 1 likewise all the datasets are built for all the layers based on their own layer priority.

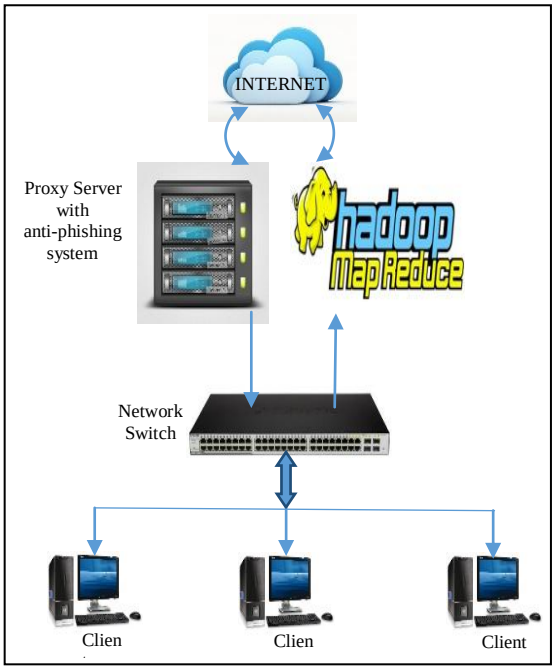
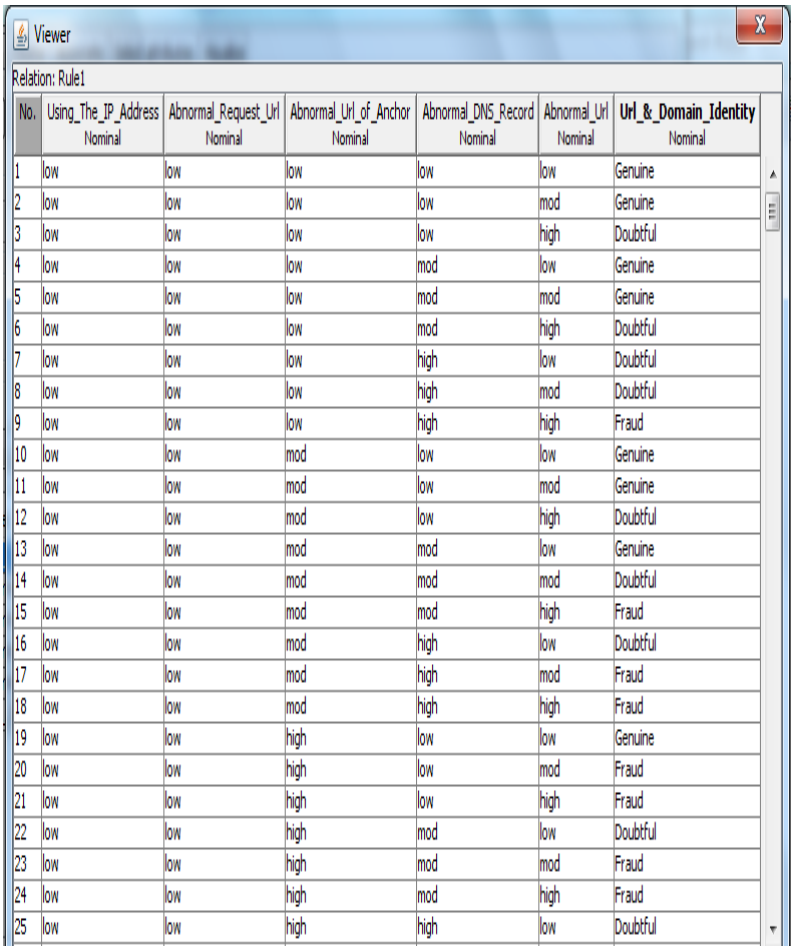


Figure 4 Proxy server-client architecture



Viewer

Relation: Rule1

No.	Using_The_IP_Address Nominal	Abnormal_Request_Url Nominal	Abnormal_Url_of_Anchor Nominal	Abnormal_DNS_Record Nominal	Abnormal_Url Nominal	Url_&_Domain_Identity Nominal
1	low	low	low	low	low	Genuine
2	low	low	low	low	mod	Genuine
3	low	low	low	low	high	Doubtful
4	low	low	low	mod	low	Genuine
5	low	low	low	mod	mod	Genuine
6	low	low	low	mod	high	Doubtful
7	low	low	low	high	low	Doubtful
8	low	low	low	high	mod	Doubtful
9	low	low	low	high	high	Fraud
10	low	low	mod	low	low	Genuine
11	low	low	mod	low	mod	Genuine
12	low	low	mod	low	high	Doubtful
13	low	low	mod	mod	low	Genuine
14	low	low	mod	mod	mod	Doubtful
15	low	low	mod	mod	high	Fraud
16	low	low	mod	high	low	Doubtful
17	low	low	mod	high	mod	Fraud
18	low	low	mod	high	high	Fraud
19	low	low	high	low	low	Genuine
20	low	low	high	low	mod	Fraud
21	low	low	high	low	high	Fraud
22	low	low	high	mod	low	Doubtful
23	low	low	high	mod	mod	Fraud
24	low	low	high	mod	high	Fraud
25	low	low	high	high	low	Doubtful

Figure 3: Sample Dataset for Layer 1

B. Experimental Setup

One can think of two possibilities of implementing our system one is at cloud services and other at proxy server [14]. For the prototype consideration we have implanted it as prototype proxy server we can say server client model. Scenario is like one proxy server handling all the incoming requests from all network components under it. At normal proxy server at peak time several thousands of URL requests might be received. Processing all these request for phishiness using sequential processing will take considerable amount of time for that reason we are accelerating process by using Hadoop-MapReduce. At proxy server each and every request received will be processed as it will be given as input to our phishing detection system. For assigning values to he attributes small JAVA codes are written which will ultimately call Hadoop-MapReduce wherever necessary. All these layers and Hadoop-MapReduce needed to be integrated by some means for that purpose we are using files as intermediate of input output mean. All the input output communication is done by using simple files. Figure (4) shows proxy server-client architecture with the facility of Hadoop at server.

VII. EXPERIMENTAL RESULTS

One obvious result of using the Hadoop-MapReduce is speedup. For results we are going to compare the proposed system against existing all type of anti-phishing systems using timely updated lists, Toolbars, data mining based systems. There are data mining based anti-phishing systems which use browser extensions just to send URL at processing site and then process URL at processing site ad send back result to the requested user. This process obviously take several seconds thought it takes several milliseconds for actual processing. It suffers from network propagation delay. Instead we brought the processing at proxy server so it that delay is minimized. Nevertheless Hadoop-MapReduce gives another push to the speedup. Graph fig (5) shows result of comparison of proposed system with all existing anti-phishing systems. As the system is placed at client location itself response time of the system is less than the other toolbars or browser Add-on. Browser Add-on and toolbars in average take response time in seconds but our proposed system takes response time in milliseconds only. One thing to be noticed is when the URL is genuine it takes a lot of time for response from other systems as it has to go through all processing to prove it is Normal URL so the response time increases but in case of our system that time is also reduced though it has to go through extensive processing. Moreover in over system we are processing hundreds of URLs at same time whereas all other systems process only one URL at a time so there is no comparison as such.

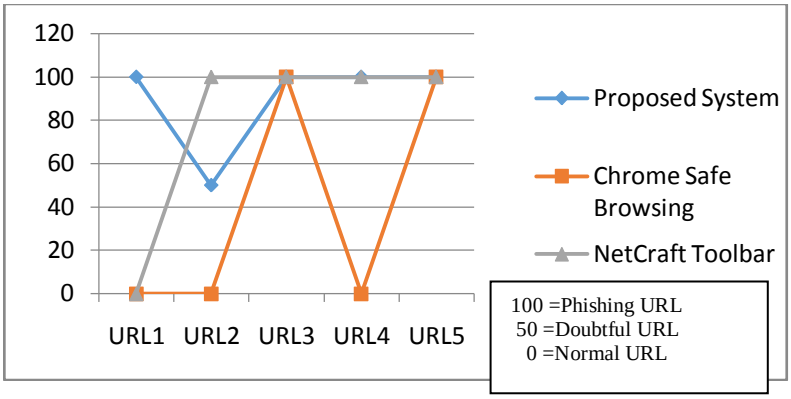


Figure 5 Comparison of randomly selected URLs

VIII. CONCLUSIONS AND FUTURE WORK

Main goal of the system is to achieve speed up in existing anti-phishing system by some means. Using Hadoop-MapReduce in integration with anti-phishing technique we have achieved considerable time speedup. Even if the phishing webpage is not showing phishing characteristics very clearly at first layer it might show characteristics in the next layer so that no phishing webpage will pass through our system. This is the advantage of having layered architecture of attributes. Hadoop-MapReduce will increase the response time of the system considerably. This system is very effective in securing network from phishing attach even at its best. There is a lot of scope for improvement of this system. One can improve the performance of system by converting this as a cloud service. As per type of organization we are protecting from phishing attach change the attributes to be considered for making effective decision about the phishiness of the system.

REFERENCES

[1] http://www.netcraft.com/anti-phishing/.
[2] http://windows.microsoft.com/en-in/windows-vista/phishing-filter-frequently-asked-questions.
[3] http://www.google.com/tools/firefox/safebrowsing/.
[4] docs.apwg.org/reports/apwg_trends_report_Q4_2012.pdf.
[5] http://apwg.org/sponsor-solutions/research-partners/.
[6] http://www.phishtank.com/friends.php
[7] Fu, A.Y.; Liu Wenyin; Xiaotie Deng, "Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)," Dependable and Secure Computing, IEEE Transactions on , vol.3, no.4, pp.301,311, Oct.-Dec. 2006.
[8] Hong Bo; Wang Wei; Wang Liming; Geng Guanggang; Xiao Yali; Li Xiaodong; Mao Wei, "A Hybrid System to Find & Fight Phishing Attacks Actively," Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on , vol.1, no., pp.506,509, 22-27 Aug. 2011.
[9] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM - 50th anniversary issue: 1958 - 2008, vol. 51, no. 1, pp. 107-113 , 2008.
[10] http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/.
[11] Aburrous, M.; Hossain, M.A.; Dahal, K.; Thabatah, F., "Modelling Intelligent Phishing Detection System for E-banking Using Fuzzy Data Mining," CyberWorlds, 2009. CW '09. International Conference on , vol., no., pp.265,272, 7-11 Sept. 2009.
[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
[13] http://www.cs.waikato.ac.nz/ml/weka/documentation.html.
[14] Tanenbaum, A. S. (2010). Computer Networks (5th Edition). Prentice Hall; 5 edition (October 7, 2010).

