ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse243952020.pdf
https://doi.org/10.30534/ijatcse/2020/243952020

Detection of Malicious Web Pages Using Machine Learning Technique

Ihedioha Uchechi Michael[1], Taoufik Elmissaoui[2], Okereke G.E [3], Chikodili H.Ugwuishiwu[4] Nnamdi Johnson Ezeora[5], Obayi Adaora Angela[6]

[1]Department of Computer Science, University of Nigeria, Nsukka (UNN), Uchechi.Ihedioha@unn.edu.ng
[2] Carthage University, SUP'COM, Innov'com Lab & Higher institute of Applied Mathematics and Computer Science, University of Kairouan, Tunisia, elmissaoui.enit@gmail.com,
[3]Department of Computer Science, University of Nigeria, Nsukka (UNN), george.okereke@unn.edu.ng
[4]Department of Computer Science, University of Nigeria, Nsukka (UNN), chikodili.ugwuishiwu@unn.edu.ng
[5]Department of Computer Science, University of Nigeria, Nsukka (UNN), nnamdi.ezeora@unn.edu.ng
[6] Department of Computer Science, University of Nigeria, Nsukka (UNN), adaora.obayi@unn.edu.ng

ABSTRACT
The Internet is used as a convenient channel for the distribution of information and resource sharing. This platform is made efficient and accessible by the use of search engines (electronic library) to effectively satisfy the needs of various users. Most URL are dishonest and sometimes position itself at the top of the engines. This work detects such malicious URLs using a built machine learning which we implemented to organize Uniform Resource Locator (URL) into two categories - trustworthy and untrusted. This Classifier will calculate a trust score for each user in a particular URL as to ascertain the trustworthiness, which will invariably determine the safety of web pages. This trust score will be used to decide a trusted webpage, should give accuracy of not less than 98.9% and a reasonable feature measure. The dataset are upload to the application to build model deployed from a general framework for malicious URL detection in order to predict, classify a URL and was implemented using naïve Bayes. And the URL is entered and click on Detect & Analyze and the result is displayed. This trust score of the feature extract will be used to confirm if an Internet link is trusted or not.

Key words: Social Reverse Engineering (SRE), Naïve Bayesian Classifier, Social Network Sites, Malicious Web Pages, Machine Learning

1.INTRODUCTION

Spamming is one of the major challenges facing Internet users today. Spamming is normally done by sending inappropriate or unwanted messages to a number of users over the Internet including Social Network Sites (SNS) in areas such as advertising, phishing, etc. Phishing attacks cost lots of fortune in cash and resources to business organizations and end users.

The innocent users submits information with the thoughts that the websites is a trusted source. Unfortunately, there is a high rate of increase on cybercrime with a lot of complexity. Cybercrime attacks include online frauds, cracking into the system, phishing attacks [1]. Initially, spam attacks focuses more on email which was a key communication tool via the Internet. It was easy to harvest or generate Email addresses from sources like: chat rooms, websites, customer lists and user's address book. Somehow, email filters became more advanced and this has successfully and effectively minimized the clogging of spam in email inboxes. It is obvious that spammers have advanced into a new SNS, drastically spreading malicious URL through fake accounts, Instagram, Bulk messaging. The problem of analyzing one by one makes malicious URL of great concern. [2]

Malicious URLs mislead or hack users and their network. It is therefore imperative to study and develop procedures that will detect malicious web pages, which has become another worry of the century in cyber security.

SR in software engineering shows that the internet is saddled with security risks from hackers, spammers who daily seize this opportunities to swindle fortune from both individual and corporate entities through the spread of malicious web links [3]. There are some problems in the existing system are which include: Lack of improper security systems for internet users, Lack integrity on online transaction, the porosity of the internet pave a way for internet hacking.

This work designed a machine Leaning Techniques for detecting malicious web pages using the Naïve Bayes machine learning model (Algorithm).

The achievement of this work are set objectives like, Show brief web analysis on URL (Long and Short) either a trustworthy or trusted web page, to determine untrusted (web spam) using machine learning techniques Naive Bayes, develop a structural model for the system implementation and to create a web based application for detecting and tracking a malicious URL both long and short. Google Scholar as used in this work has been identified as included in Cochrane and JAMA systematic reviews [4] this alone does not justify this method as it can be criticized as well.

The exponential connectivity to Internet connectivity in devices such as laptops, smart phones and many other electronic devices enhances users to share interconnectivity, however, and this is a major medium phishers target to carry out Phishing activities [5].

2.THEORETICAL BACKGROUND

The engagement in this literature search was done in adoption of [4] using some identified search string, so therefore we executed the search using the following search strategy like:

1. The automatic searching in proven databases for literatures which includes: IEEE Explore Digital Library, Science Direct and Google Scholars.
2. Manual search in conferences proceedings and journals
3. Collections from set of Spam detection papers
4. Search period: inception of internet –2019

2.1 Related Works

Phishing is constantly growing since it is stress-free to copy the totality of any website using the HTML source code. This further specifies that approaching the detection of malicious URL will require analyzing html feature. Recent CAPTCHA authentication application designed in an inexpensive mode protects the security of vulnerable users by enabling safe online account authentication, thereby addressing the online threats. The major challenge is in solving the general ignorance of safety warning as well and trading authentication on conceded hosts of a secure and healthy online transaction system. There are some proposed hardware solutions but these are not are not realistic for the home users due to its outrageous cost.

In their work [5] it was cited that there are also more than a few viable digital forensics software suites available for probing computer crimes.

In [6] they established an email alert detection system which focused on detection of deceptive e-mail. However, their influence gave 88.6% accuracy in detection. This and lots more drive our motivation to do a further study involving the phishing detection and prevention covering both client-side techniques and server-side techniques.

[7] In their work invented a method of active blacklisting which used inference domain figured out from by those exact fraudsters whose domains are part of the confirmed blacklists schedule. WHOIS and DNS assist in capturing the domain information. However, their performance gave a yield of domain inference of 73% in the blacklists later on. This proactive technique updates attacks with the already existing blacklist domains kernel making it unreliable for skilled new attack inventions that are different form the existent ones.

[8] A fascinating work which developed a system that is able to capture any malicious URL domain by training the classifier, the domain relationship not minding. This method achieves a good accuracy for phishing detection, indicating the effectiveness of the proposed mechanism. The previous study that is analyzed, on the other hand, exposes a performance decline due to the evolution of the phishing ecosystem, while our proposed methodology and feature set demonstrates significant superiority. Nevertheless, there are some limitations are one, that all the data set examples used were downloaded from single sources, two, though no bias das result of data loss. We require further arrangement and participation with a wider range sample with more complete data. Besides, this work did not consider other features used by the current machine learning process. Therefore, they did not focus on analysis to understand the classification process and reduce features to minimal as to arrive at a similar result.

[9] In their research studied a comprehensive list of features comprising from X.509 certificates. They used of Alexa ranking of top one hundred websites and malicious URLs which were collected from PhishTank as the data set. This record known accuracy of 93.7% to 95.5% the limitation is in the restrictive use of feature set of certificates leaving out other attributes like network performance internet feature etc.

[10] The work studied multi-layer machine learning detection called CANTINA+, Features extracted from URL, HTML DOM, PageRank are used for detection technique. We discuss this in similar to the work.

[11] Worked a similar yet another framework of HTML DOM based approach for the detection of malicious URL. Weak points drawn from this method can be easily be manipulated by to deceive the defeat the classification system. Since the classification page content will be downloaded first, the malicious attribute can be introduced and downloaded to the user machine same time if the method is done using the user's browser. It is also identified that cloaking technology can beat these methods. By this, we see different drawback of the few mentioned. [12] In their work proposed linked and context based techniques for automating the detection of Web spams.

[13] Proposed a work to classify the abnormalities in ECG signals through Signal Processing and Feature Extraction from the Time series dataset using. The work being a combination of Multivariate Maximal Time Series Motif with Naïve Bayes Classifier classifies the ECG abnormalities in absence of prior knowledge. It achieved the accuracy of 93.33% and is 98% than the existing techniques. This also supports the choice for Naïve Bayes Classifier in our proposed system.

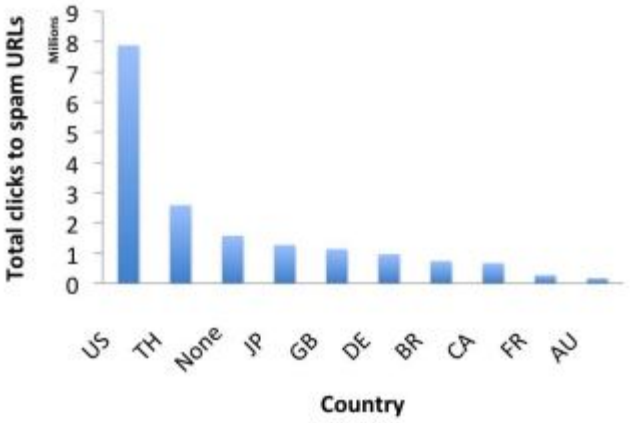
[14] Proposed URLNet, which is an end-to-end deep learning design. The model is allowed to take a number of semantic information which was not easy for other models studied thus far.

[15] This work focused on twitter account in order detect spam shortened URL on twitter. There data collection was done through use of APIs: Twitter APIs and Bitly APIs. They proposed a Random Tree algorithm. Refer to tales 2.1, 2.2 and fig 2.1, 2.2.

Top-10 short URL providers in the Twitter dataset

Short URL Provider	*Count*
Bit.ly	641,423
t.co	129,677
Tiny.com	62,488
Ow.ly	42,838
Is.gd	14,664
Goo.gl	13,122
j.mp	8,963
Su.pr	3,764
Twurl.nl	2,807
Migre.me	2,788

Table 2.1 Twitter Dataset (source:[15])



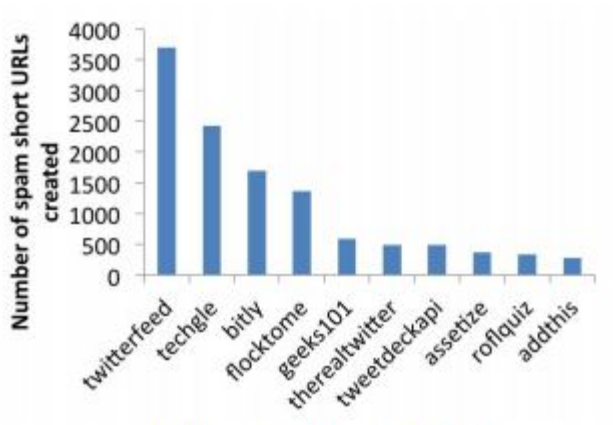
Top-10 countries based on clicks to spam URLs

Figure 2.1 Spam URL (source: [15])

Top-10 creators that created only spam short URLs

Creator	*Spam URLs / All URLs*
dailypiff187	150/150
golfhonchonow	72/72
headlinehoncho	63/63
newswatchphilly	56/56
mskaya4u	56/56
golfhonchotoo	50/50
golfhoncho	48/48
breakingnewssource	47/47
onlinenewsblast	47/47
portlandtimestribune	46/46

Table 2.2 Sapm Short URLs [15]



Top-10 creators of spam short URLs

Figure 2.2 Spam Short URLs Chatt [15]

It is necessary that we detect and act on such threats in an apt routine. The existence of short URL creates

a gap and question on how it started. Simple and short, setting limitation in characters in SNSs motivated users to use shorter services such as we see in tweeter with between 140 characters maximum.

[16] In their investigation elucidated about the safekeeping apprehension in economic viewpoint how these malevolent substances have wedged our financial resources. The motives are not far from the resulting:

i. The negative effect on the overall public which ascends when procedures are openly shelter off and communicated on
ii. Selected entitlement that revelation of material about the crime victims actually aid Attackers more than it helps guards.

[17] The novel work found out that weakness in an open source software could be frequently exploited much more in a regular mode than in closed software. But there are also some other clear advantages to the public than this wicked occurrences.

[18] Scrutinized the features of the email by the use of HTML tag, number of links and by means of (support vector machine) SVM.

[19] They had their way of contribution by the gathering more than two hundred and thirteen thousand online users who cooperated on Koobface. They engaged the internet sites for more than four days in order to blacklist but only 27% of URLs were unbearable and 81% users clicked on Koobface spam.

As used in [20] decoy profile approach to gather social evidence from SNS feature. These are used to identify spamming in social network without considering the malicious link or content posted. Results show spammers sends several friend appeal but they are not a real-life friends.

In [21] an algorithm was developed for scheming trust score for each user in mixed SNS. The research work gave feature amount of 81% and an accuracy of 92.6%. Understanding long and Short.

In [22] their work stated that shortening URL is basically to reduce long URL. Social network will never know which website to assailant who created malicious URL prefers to post short URLs. They create a good affiliation with any users by sending valid short URLs. When they generate the trust then they start directing malicious links. User trust the link that is posted.

2.2.1 Machine Learning Framework:

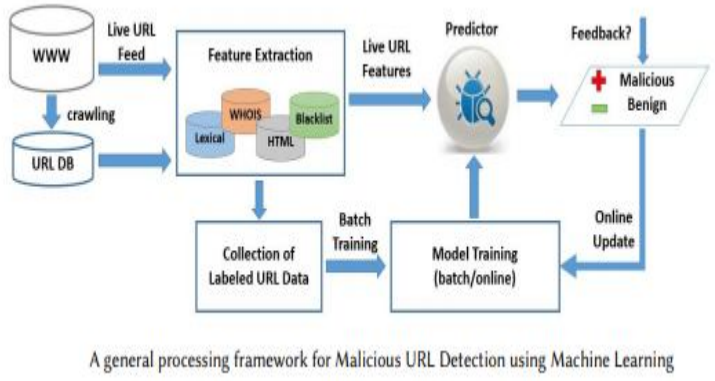


Figure 2.3 Malicious URL Detection [23]

Refer to figure 2.3, the approach in [23] analyzes, extracts the very good features and synthesis URL. They outline in clear statements depictions of URLs and prediction model on data of both malicious and benign. Static features, and dynamic features can be applied. In static analysis, analysis of a webpage based on information available is performed without executing the URL (examples are executing JavaScript or other code) [24].

3.INTRODUCTION

This part specifically involve looking at a system, determining how well it functions, specifically changes that needs to be made and the quality or impact of the output that will be achieved. Designing of a system involves the integration of various procedures into the subsystems components and articulating them together into the main system in order to achieve the expected goal. Analysis and designing of a new system requires modeling methodology that creates a representation of the real world entities of a system [25].

3.1 Detecting Malicious Web Pages

The growth and advancement of the internet technology and most victims are most of the time custom, the end users. The dataset collected are sets of URL indicating if the link is trusted or not. The platform is designed to detect a malicious web address either long or short. The dataset source is the kaggle.com. The normal URL of the dataset is labeled and the malicious URL of the dataset are labeled. From the beginning, it was identified to be

trustworthy and untrusted web links and pages (Short and long).

3.2 Analysis of the Existing System (Baseline System)

Internet is a necessary but transports stern data security challenge [26]. The installation of antivirus has not totally eradicated attack of malicious web pages because users are still hunted carelessly by malicious web pages, such as webpage viruses, spam, phishing sites[26][27][28].

Malicious web page detection becomes the objective of the project work to avoid the attack from the hijackers or even customers as a victim. The baseline system through feasibility checks on existing system could actually detect but could not predict the occurrence of malicious webpages. The system is still a conventional i.e using a parameter to determine malicious webpage link. The system still static in nature through presenting parameter with value to determine the presence of malicious webpages link.

Every old system has its failure that led to the proposed work after the feasibility study that reveal the problems of the existing system. This actual emphasis on the failures of other methods of classification. These were the untrusted and trustworthy is classified based curative when the attack has already taken and made damages to victim system. The following forms the shortcoming as;

1. Remedial measures to prevent an attack by the fraud stars.
2. It forms an open door to fraud start rights to piracies
3. It makes untrusted site not to be visited

The system is very fragile and allow unauthorized users to have access to private properties thereby making the internet users victim of fraud starts. The system is very porous and cannot protect the users from the intending attack from hackers. Malicious web pages affect internet business and transaction, this is whereby the internet users are very fearful of using the internet as a facility for business dealings. The following are the weakness of the system;

1. sites are not trusted for online payment system
2. it lack privacy and protection on credit card details
3. inability to analysis features potentially indicative of suspicious URLs

3.3 Proposed system

The existing system, after a proper feasibility study it was observed that the traditional system require a change from its conventional method to machine learning approach to malicious. The proposed system was motivated based on the failures of the existing system and the porous internet web system. The following are the reason for proposing the new system;

1. insecure system for online internet transactions
2. unprotected data on the internet privacy
3. credit card details not secured on internet platforms for example SSL

A machine learning algorithm is proposed for professional classification of web link in to trusted and trustworthy link. The Naïve Bayes classification algorithm uses a probabilistic pattern of prediction to determine the actual outcome of a given statistical analysis of URLs. The proposed system is an advance method of malicious web page classification i.e separating the trustworthy or untrusted spam pages. The internet users are fraud through this vulnerable pages thereby put more fear in the heart of internet users. The system provide this advancement of malicious internet web pages;

1. Machine learning approach to spam web page classification
2. The use of pre-existing data to actualize the prediction
3. The system is a preventive system to avoid users stumbling into a malicious web site.
4. The vulnerable system assist the fraud stars to launch attack.
5. Classification algorithm to be used here is the Naïve Bayes.

3.3.1 Diagrammatical Description of the Proposed System

This section of the research work absorbed the block diagram of the entire system in figure 3.1. These describe the functional object within the centralize certificate verification and authentication.

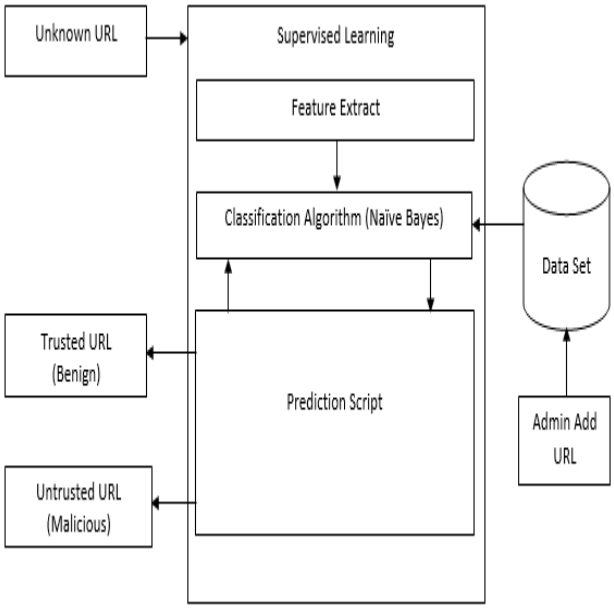


Figure 3.1 Block Diagram of Malicious URL

Unknown URL: this a suspected URL which passes through machine learning algorithm.
Data Set: is an excel file with an extension csv where the URL are stored.
Classification Algorithm: The classification algorithm processes the features to crop a class decision or diagnosis.
Admin Add URL: The Admin is responsible updating new suspected URL
Trusted URL: known as the Benign, this are the trustworthy link which is not harmful
Untrusted URL: the known as the malicious, this are untrusted link which is harmful

3.2.2 Operation Process of the Diagram

The project work is designed and implemented to filter URL. The system is implemented using a classifier algorithm. The program is implemented in python language, the framework is flask which is light weight micro outline designed to grow application from simple to factory application. The system have an input text-box for where unknown URL is entered to system for verification/classification. The URL passes through a machine learning algorithm known as the naïve bayes classifier, the system classifies it base on trustworthy or untrusted web link. The accuracy of prediction is on numbers of URL classified based on trained and test dataset. The naïve bayes uses a classifier procedure that is built on probabilistic theory that presents result as 1 or 0, true or false etc. Other open source software have its bayeness in predictions i.e our system update website based on the fact that any new malicious web is found, the characteristic is update to avoid its type to load on our Computer system.

3.3 Comparing Existing and the Proposed System

The comparing of the existing system and the proposed system as seen in table 3.1

Conventional Malicious Web Detection	Machine Learning Malicious Web Detection
1. Human determined malicious web link	1. Machine predicted malicious web pages
2. Not real-time responsive	2. Realtime instant response in prediction
3. The system is static in nature	3. The system is dynamic nature
4. The platform is not distributive	4. The system is accessed on the world wide web

Table 3.1 Existing System versus Proposed System

3.4 Malicious Search Modeled Algorithm

Search algorithms form an important part of many programs. The following are the search algorithm:

- Linear search
- Binary Search
- Tree Search
- Genetic Search

3.5 Diagram Representation (Unified Modeling Language)

It is the unique system of diagrammatically representation of the complex and complete system. The system is designed and implemented using the python flask framework. Several UML diagram are used to explain the proposed system.
The system backend is completely as Naïve Bayes probabilistic algorithm classifier. The naïve Bayes statistical techniques converted into python program. The frontend application is a bootstrapped user interface that describes the request entry point of the system through the URL. The system classifies the URL based on Benjng or malicious.

3.5.1 Class Diagram

This UML class diagram is used to represent the structural class documentation of the system. The UML diagram uses a classification pattern which work well with Naïve Bayes algorithm to prediction URL either benjng or macilious page link. This link are very harmful to internet users. Through this harmful web link the customer could be defrauded or hijack very confidential document. This is shown is figure 3.2.

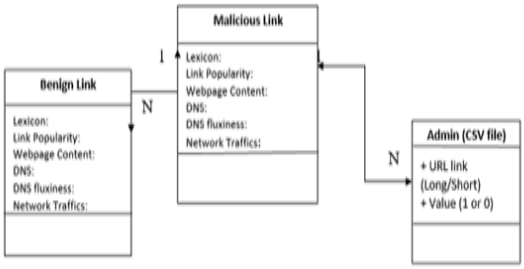


Figure 3.2 Class Diagram of the Proposed System

The new system to replace the existing system has the above class diagram. The class diagram describe each functional classes of the proposed system. Through the above classification algorithm the URL is filtered.

3.5.1 Use Case Diagram

The figure 3.3 is a design model to demonstrate the different ways that a user might interact with a system. The classification algorithm is for classification of URL. The system determine malicious link by raising alarm to indicate intrusion.

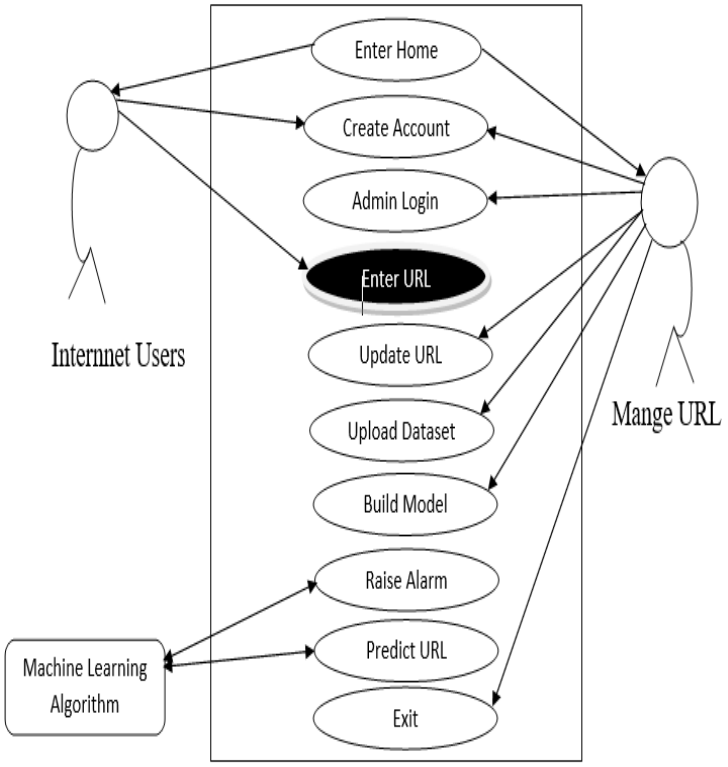


Figure 3.3 Use Case Diagram

The client users are privileged to create account and login to have access to the machine for prediction. The users enter the URL and click predict to determine if the web link is harmful. The admin is responsible for update URL i.e making the system to be dynamic by adding link into CSV file. The machine raises alarm once a link is malicious is found.

3.5.2 Sequence Diagram

This shows communication among objects in a sequential.

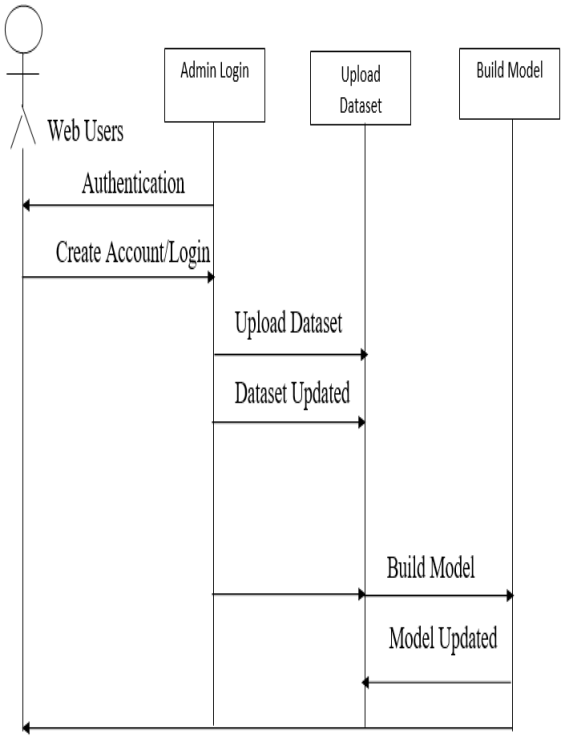


Figure 3.4a Sequence Diagram of the Admin Task
Administrator is responsive for model building and uploading dataset. Once the dataset are update or a new dataset is required for building a new model the administrator does the process to predicting whether trustworthy or untrusted.

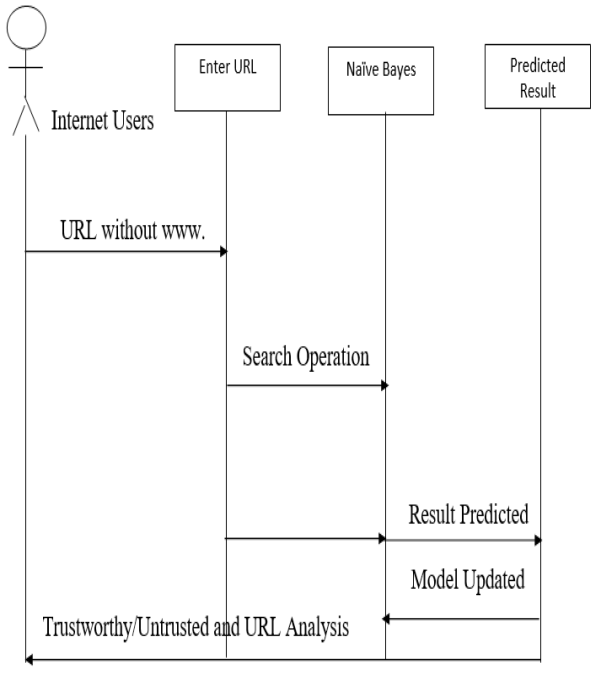


Figure 3.4b Sequence Diagram of the Internet User Task

The sequence diagram, figure 3.4a and 3.4b describes the process of users predicting trustworthy or untrusted URL. The web users are create account, and login to have access to the prediction machine. A suspected web link is entered into the URL Request and click on prediction to display the result. The admin update URL on the dataset.

4.SYSTEM IMPLEMENTATION

In this research work, python flask framework was installed on a personal computer and the software was coded in a notepad++ (text editor). The platform was deployed locally on the system for testing. The coding involves a systematic representation of the implementation model like spiral and Object Oriented Analysis and Design. The implementation was actually coded in a software development environment. The entire system was tested and debugged through which errors are eradication from the program/software at different stages of the development. Like stated before the software was tested in a local server for efficient execution.

4.1 Choice of Development Environment

Every research work has it preferred language to be used for implementation. For example implementation of an enterprise web based application which demand a unique web framework like python flask etc., standalone application which can implemented using the vb.net framework to actualize it. So most of the time choice of language depends on the area/field of application. The project work "**detection of malicious web pages - a context based approach using machine learning technique**" requires a web based system access and distributive at real time responsive manner. The system been a web based platform is designed to respond to real time request of URL detection or malicious page used by the hackers and fraud star to trap credit card details on the internet. The project work is target to reducing and avoid fraudulent activities by cybercrime persons. The machine learning techniques are implemented in python. The datasets are classified and labeled. The administrator is required to upload dataset in CSV extension file and build model. And the internet user enters the URL without www. To determine if trustworthy or untrusted with URL analysis.

4.2 Implementation Requirement: Machine Learning Algorithm for Naïve Bayes

Given, n different attribute values, the likelihood now can be written as

$$P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

Here, X is attributes or features, and Y is the response variable.

Now, P (X|Y) becomes equal to the products of probability distribution of each attribute X given Y.

Python Interpretation of Naïve Bayes Algorithm

```
# load the iris dataset
from sklearn.datasets import load_iris
iris = load_iris()

# store the feature matrix (X) and response vector (y)
X = iris.data
y = iris.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=1)

# training the model on training set
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(X_test)

# comparing actual response values (y_test) with predicted response values (y_pred)
from sklearn import metrics
print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy_score(y_test, y_pred)*100)
```

4.3 Our System Architecture

Selection of a right implementation method will deliver progressive outcomes, therefore in figure 4.1, we employ the use of iterative incremental approach.

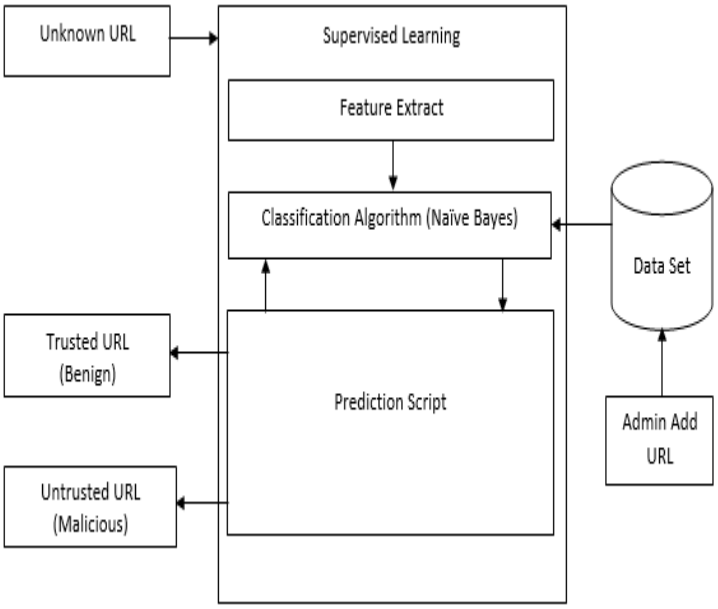


Figure 4.1 Implementation Architecture

4.4 Actual Test Result versus Expected Test Result

After carrying out a test with the recent URL dataset collected as an input data – information details of student for several universities based on random selections. The testing process was initialized using the predominant data of a view student collected from the selected institutions, referring to tale 4.1.

TEST DATA	ACTUAL TEST RESULT	EXPECTED TEST RESULT
Malicious, Benign URL	 The above result of the system describe the actual outcome the process, the Youtube.com as the input is a trustworthy URL and the right hand side become the analysis of the URL	The model to predict the URL entered either a trustworthy or untrusted URL and analysis

Table 4.1 Result Table using Test data.

Figure 4.2 up to figure 4.5 shows some Program Snapshots with are labeled according to there meaning

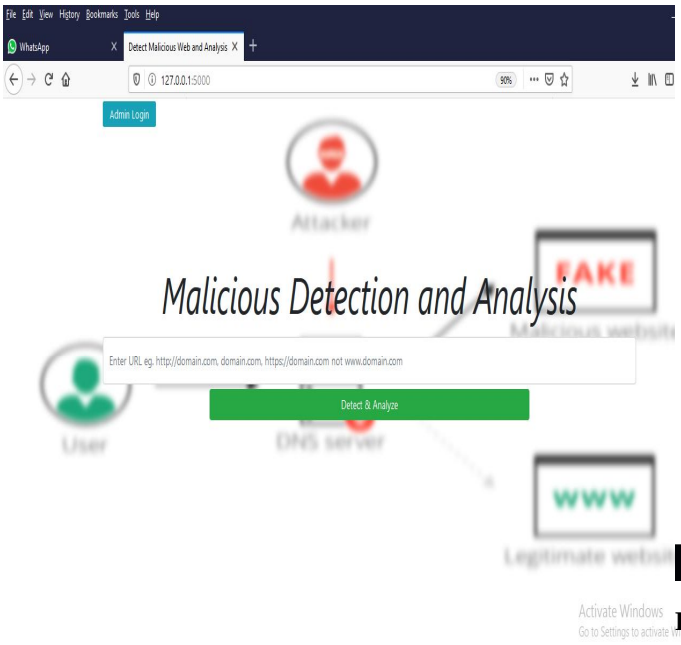


Figure 4.2 Index Search Page

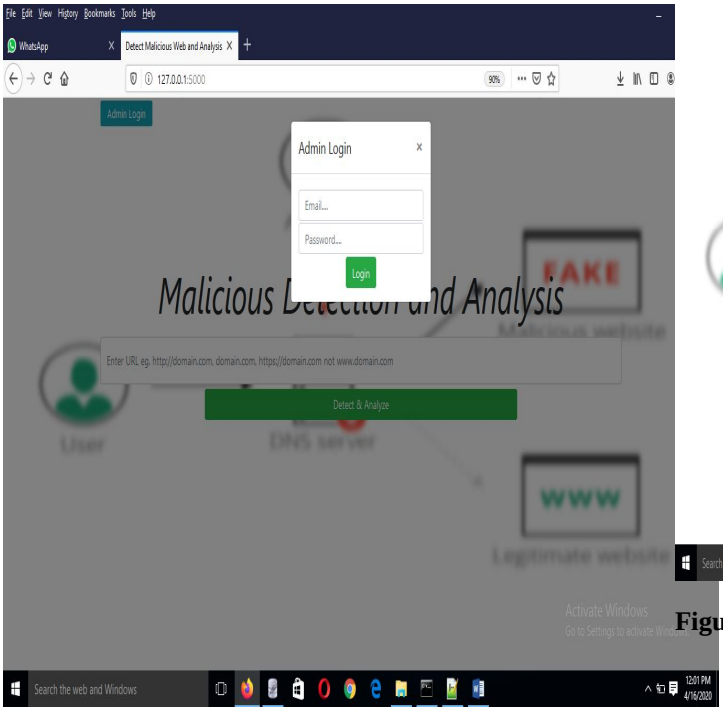


Figure 4.3 Admin Login Page

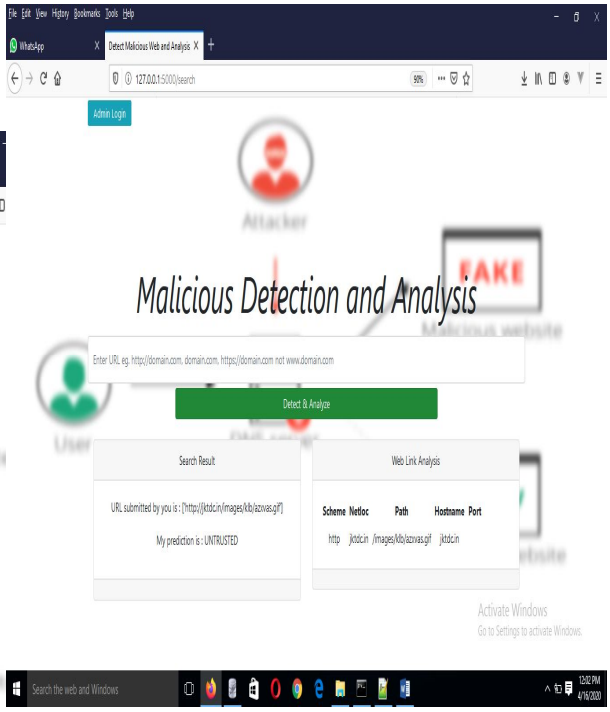


Figure 4.4 Untrusted Page

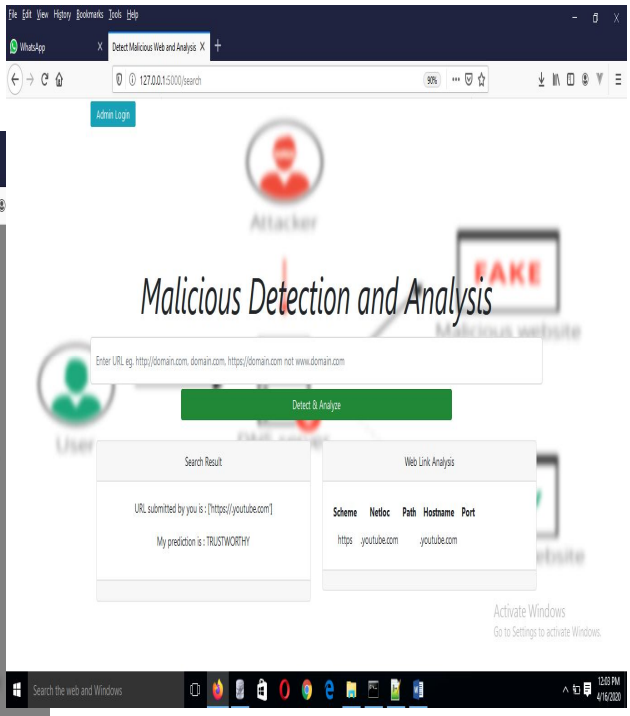


Figure 4.5 Trustworthy Page

4.5 Dataset We used a range of online downloadable dataset, the figure 4.6 shows the insight as used.

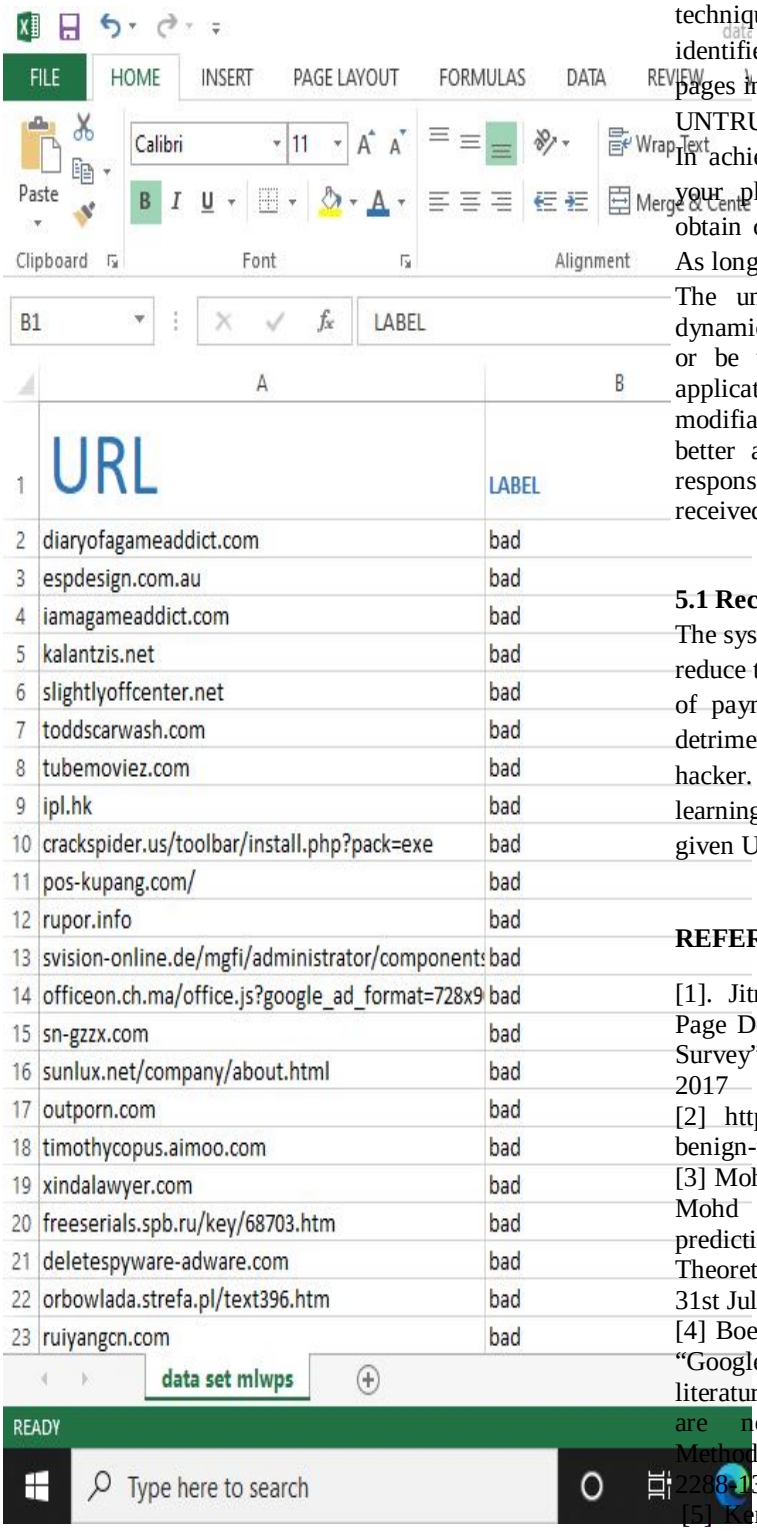


Figure 4.6: Data set for Modeling [extracted online from 2]

5. CONCLUSION

The work "detection of malicious web pages - a context based approach using machine learning technique" detects malicious pages. This has been identified as a machine learning task. We categorize pages into two groups as either TRUSTWORTHY or UNTRUSTED.

In achievement, transaction alerts will be raised on your phone ones the fraudsters clone website to obtain credit card and other payment source details. As long as this app is installed on your phone.

The uniqueness of this research work (app) is dynamic but other are static, the dataset could change or be updated to improve the prediction of the application. Models in form pickets are also modifiable and it also build malicious models for better accuracy of prediction. We have real time response system, result are generated once input is received with the URL Link analysis

5.1 Recommendation

The system is suggested for cooperate organization to reduce the level of risk applied to the online activities of payment using financial detail which could be detrimental having it in the hand of fraud star or hacker. The entire system is attached to a machine learning algorithm which requires for predicting a given URL (Trustworthy or Untrusted web link).

REFERENCES

[1]. Jitndra Agrawal, etal, 2017, "Malicious Web Page Detection through Classification Technique: A Survey" IJCST Vol. 8, Issue 1, page 1, Jan - March 2017

[2] https://www.kaggle.com/xwolf12/malicious-and-benign-websites

[3] Mohd N.M, Suriayati C, Anusuyah S, Aswami F, Mohd Z, Mohammad Z, Fakhrul A " Malware prediction algorithm: Systematic review" Journal of Theoretical and Applied Information Technology 31st July 2018. Vol.96. No 14

[4] Boeker, M., Vach, W., & Motschall, E. (2013). "Google Scholar as replacement for systematic literature searches: good relative recall and precision are not enough". BMC Medical Research Methodology, 13, 131.http://doi.org/10.1186/1471-2288-13-131

[5] Kenneth Fon Mbah, 2017, "A Phishing E-Mail Detection Approach Using Machine Learning Techniques" A Thesis Submitted in Partial Fulfillment of The Requirements for Master of Computer Science in the Graduate Academic Unit of

Faculty of Computer Science, UNB, the The University of New Brunswick, https://unbscholar.lib.unb.ca/islandora/object/unbscholar%3A8107/datastream/PDF/download/citation.pdf
[6] Yoshiro Fukushima, Yoshiaki Hori, and Kouichi Sakurai, 2011, "Proactive blacklisting for malicious web sites by reputation evaluation based on domain and ip address registration", In Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, pages 352–361. IEEE, 2011
[7] YAOKAI YANG, 2019 "Effective Phishing Detection using Machine Learning Approach" online at
https://etd.ohiolink.edu/!etd.send_file?accession=case1544189633297122&disposition=inline
[8] Zheng Dong, Apu Kapadia, Jim Blythe, and L Jean Camp, 2015, "Beyond the lock icon: real-time detection of phishing websites using public key certificates". In Electronic Crime Research (eCrime), 2015 APWG Symposium on, pages 1–12. IEEE, 2015.
[9] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. "Cantina+: A featurerich machine learning framework for detecting phishing web sites". ACM Trans. Inf. Syst. Secur., 14(2):21:1–21:28, September 2011.
[10] Angelo PE Rosiello, Engin Kirda, Fabrizio Ferrandi, et al, 2007, "A layout-similarity-basedapproach for detecting phishing pages". In Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on, pages 454–463. IEEE, 2007.
[11] Vikash Kumar Singh 2009, "Machine Learning Techniques for Detecting Untrusted Pages on the Web", Online at https://pdfs.semanticscholar.org/1fed/02fb3fd318390f58d6bbbc0fcfe153e79fb1.pdf
[12] S. Padmavathia, E. Ramanujamb, 2015, "Naïve Bayes Classifier for ECG abnormalities using Multivariate Maximal Time Series Motif", ScienceDirect, Procedia Computer Science 47 (2015) 222 – 228
[13] Hung Le, Quang Pham, Doyen Sahoo, Steven C.H. Hoi. 2018. "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection". In Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17), 13 pages.https://doi.org/10.475/123_4
[14] Wang, D., Navathe, S., Liu, L., Irani, D., Tamersoy, A., & Pu, C. (2013). Click Traffic Analysis of Short URL Spam on Twitter. Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. doi:10.4108/icst.collaboratecom.2013.254084
[15] https://developer.twitter.com/ Assessed Online
[16] Anderson, R., & Moore, T. 2006, "The Economics of Information Security". Science, 314(5799), 610–613. doi:10.1126/science.1130992
[17] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. 2010, "Lexical feature-based phishing URL detection using online learning". In Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, pages 54–60. ACM, 2010.
[18] Kurt Thomas, 2010, "The koobface botnet and the rise of social malware. 2010".
[19] Ralph Gross and Alessandro Acquisti, 2005, "Information revelation and privacy in online social networks". In Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pages 71–80. ACM, 2005.
[20] Rasula Venkatesh, 2015, "Malicious Accounts Detection based on Short URLs in Twitter, https://core.ac.uk/download/pdf/80148593.pdf
[21] Dolvara Gunatilaka, 2011, "A survey of privacy and security issues in social networks", in Proceedings of the 27th IEEE International Conference on Computer Communications. Washington: IEEE Computer Society, 2011.
[22] Demetris Antoniades etal. "we.b: The web of short URLs "March 28–April 1, 2011, Hyderabad, India.
https://dl.acm.org/doi/pdf/10.1145/1963405.1963505?download=true
[23] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi. 2019. Malicious URL Detection using Machine Learning: A Survey. 1, 1 (August 2019), 37 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn
[24] Birhanu Eshete, Adolfo Villafiorita, and Komminist Weldemariam. 2013. "Binspect: Holistic analysis and detection ofmalicious web pages" In Security and Privacy in Communication Networks. Springer.
[25] Boyd, D., Ellison, N., 2007, "Social network sites: Definition, history, and scholarship" Journal of ComputerMediated Communication, Oct 2007, Vol 13, no 1, pp. 210-230
[26] http://topsitesblog.com/list-of-social-networking-sites/
[27] Rashmi A.Zilpelwar, Rajneeshkaur K.Bedi, Vijay M.Wadhai, 2012, "An Overview of Privacy and Security in SNS "International Journal of P2P Network Trends and Technology (IJPTT) - Volume 2 Issue 1 January to February 2012 ISSN: 2249-2615 http://www.ijpttjournal.org Page 14
[28]Social-networking-sites-for-healthcare-medicalprofessionals,
http://medicallabtechnicianschool.org/2009/top-25