# Phishing Detection using RDF and Random Forests

Vamsee Muppavarapu, Archanaa Rajendran, and Shriram Vasudevan
Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham University, India

**Abstract:** *Phishing is one of the major threats in this internet era. Phishing is a smart process where a legitimate website is cloned and victims are lured to the fake website to provide their personal as well as confidential information, sometimes it proves to be costly. Though most of the websites will give a disclaimer warning to the users about phishing, users tend to neglect it. It is not a fully responsible action by the websites also and there is not much that the websites could really do about it. Since phishing has been in persistence for a long time, many approaches have been proposed in past that can detect phishing websites but very few or none of them detect the target websites for these phishing attacks, accurately. Our proposed method is novel and an extension to our previous work, where we identify phishing websites using a combined approach by constructing Resource Description Framework (RDF) models and using ensemble learning algorithms for the classification of websites. Our approach uses supervised learning techniques to train our system. This approach has a promising true positive rate of 98.8%, which is definitely appreciable. As we have used random forest classifier that can handle missing values in dataset, we were able to reduce the false positive rate of the system to an extent of 1.5%. As our system explores the strength of RDF and ensemble learning methods and both these approaches work hand in hand, a highly promising accuracy rate of 98.68% is achieved.*

**Keywords:** *Phishing, ensemble learning, RDF models, phishing target, metadata, vocabulary, random forests.*

*Received April 22, 2015; accepted September 20, 2015*

## 1. Introduction

Phishing is a well-known act of the attackers stealing the confidential information (sometimes, money) of the users by spoofing the websites or by luring the users to visit some fake sites where they disclose their personal information open to the attackers, though done unintentionally and innocently. Though these kinds of attacks are not new to the internet era, phishing has got huge attention in the past couple of years breaking major sectors like finance, social networking, e-commerce etc. causing enormous financial loss, reason being obvious. More the business is happening in the internet, more the money revolves around, so more the chances for attacks. Victims of phishing attacks always find their personal or financial information stolen without the knowledge of users. Credit card details, personal banking details, personal information, security questions, login credentials etc are the target of these phishing attacks. Spear phishing is another type of phishing attack where the attackers focus on officials like military heads, company executives and try getting their confidential credentials. According to Rivest-Shamir-Adleman (RSA) security company’s report, in November 2014 [17] there are about 61,278 attacks, marking a 76% increase from October 2014. This is mainly due to the high volumes of online shopping specially during the holiday season. Based on the above mentioned figures, RSA has estimated a total global loss of $594 million in month of November.

Attack volume on different countries in January 2015 is shown in below in Figure 1, which is accounting to a loss of $453 million approximately, which is more than revenue of a multinational company, doing well in the market [16].

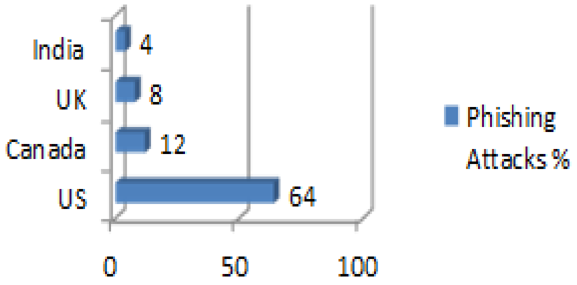

Figure 1. Statistics showing the volume of attacks in Jan-2015 globally.

The above mentioned figures can reveal someone that there is a lot of scope for designing new anti-phishing methods (techniques) that can increase the Phishing prediction accuracy and also helps in identifying the target websites of Phishing attacks.

One way to counter these phishing attacks is by creating awareness among people who use financial, e-commerce and social networking sites. Most of the current banking sites will show a phishing warning page giving instructions to the users even before they login into their account. Unfortunately most of the users neglect this, knowingly or unknowingly. Also, users may not know how to identify a phishing site technically, not all who use internet is a techie.

Therefore, there should be some mechanism that can detect the phishing pages when a user is visiting them and has to show an immediate warning about phishing attack. In literature, there are many approaches proposed to counter the phishing attacks.

These countermeasures majorly would use any of the following categories such as heuristics based approach, blacklist and whitelist approach, semantic link network based approaches and hybrid approaches.

In security domain, attackers and the people who counter the attacks always will have a “cat and rat chase”. As researchers come up with new ways to counter the attacks, attackers being smarter come up with a different way of attacking strategy. Therefore, the aforementioned approaches are not sufficient to fight phishing attacks.

This gives a room to develop a novel mechanism that uses the concepts of semantic web. Resource Description Framework (RDF) is a part of semantic web which is intended to represent the metadata about web resources. RDF provides a framework that can be used to represent the structural features and semantic features of a web page. Semantic features are the characteristics of the web page which is represented in RDF format without loss in its meaning. As structure of the RDF follows the structure of Extensible Markup Language (XML), these annotated semantic markups help the programs to understand and differentiate a webpage from other web pages, making life of researchers easy.

In this paper, we propose a novel approach that is an extension to our previous work [11]. Our approach explores the strength of RDF where we make use of the semantic features to predict the phishing web sites and their corresponding target website. In this paper, we have used a better keyword extraction algorithm as it is very important for our system. Also, we employ a better decision making algorithm that can bring down the false positives to a greater extent. As our approach depends only on the content of the suspicious webpage, we don’t need any prior data about the site. Also, this approach is capable of finding zero day phishing attacks.

The rest of the paper is organized as follows: section 2 presents the literature survey and related works in this area; section 3 discusses the architecture and information about metadata extraction and RDF generation; section 4 presents about phishing detection and target discovery; section 5 gives the performance metrics chosen and evaluation results followed by conclusion.

## 2. Literature Survey

Many anti-phishing methods have been proposed in past. All the anti-phishing methods will fall into anyone of the following categories:

1. Heuristics based anti-phishing strategies.
2. Blacklist and White list based approaches.
3. Semantic Link Network and Hybrid approaches.

Heuristic based approaches consider the characteristics of the website to predict its legitimacy. A heuristic is a feature that can be used to predict something about the behavior of the website. Chou *et al.* [4] developed Spoofguard, a browser plug-in that identifies phishing websites inspecting series of heuristics. This method uses both stateless evaluations that identify suspiciousness of the webpage extracting some features from the web page and stateful evaluations that depend on the user’s previous page visit history. This approach suffers from false alarm rate as it records some prior data about the users.

Identifying phishing emails by extracting 10 different features specific to phishing is done by Fette *et al.* [7]. Eight of these features can be extracted from the emails itself, while the other two features like age of the domain name has to be obtained from WHOIS (pronounced as the phrase who is) protocol and spam-filter output feature has to be considered to assign a class to the suspicious email. This method cannot identify pharming attacks.

One of the well-known methods in heuristics based approach is Carnegie Mellon Anti-phishing and Network Analysis Tool (CANTINA) proposed by Zhang *et al.* [22].

CANTINA is a content based approach. It uses Term Frequency-Inverse Document Frequency (TF-IDF) algorithm to extract the top ranked keywords from the page content. These extracted keywords are given as an input to a trusted search engine. In this method, a website is classified as Phishing if the page domain does not appear in the top N results of the search. Heuristics used in this approach are taken from Chou *et al.* [4] and Fette *et al.* [7] work. There is an advanced version to this method and it is called as CANTINA+ [20] where they have added ten other features including four from the CANTINA approach.

But, both these methods cannot detect phishing pages that contain more number of images and scripts. Also, they cannot detect pages if they have bad forms and bad action fields.

Pan and Ding [12] proposed an anomaly based phishing detection scheme that use the structural features of the web site and their Hypertext Transfer Protocol (HTTP) transactions. They extract Document Object Model (DOM) objects as features and applies classification techniques to come to a conclusion.

Blacklisting and whitelisting approaches are the most used techniques in the current day web browsers.

A blacklist contains a list of reported phishing pages, whereas a whitelist consists of a list of reported legitimate webpages. Whenever a user tries accessing a particular page, it is checked against the list of phishing pages available in the blacklist. This Blacklists are generally gathered from multiple data sources like spam filter, PhishTank [13] etc., Prakash *et al.* [14] used an algorithm that will divide a Uniform Resource Locator (URL) into multiple components and each component is matched against entries in blacklist. Zhang *et al.* [21] proposed a system where customized blacklists are provided for the individuals who choose to contribute data to a centralized log-sharing infrastructure. This individual blacklist is generated by combining relevance ranking score and the severity score generated for each contributor. The drawback with this scheme is blacklist or whitelist needs continuous updating mechanism. Also, the exponential growth of list imposes great deal on system resources. This scheme is not suitable to identify zero day phishing attacks as the life span of phishing page is very small.

Shahriar and Zulkernine [18] proposed a scheme that identifies suspicious web sites based on the trustworthiness testing. In this method, they check the behavior of a website with knows behaviors of phishing or legitimate sites. Based on this behavior they construct a semantic link network that is based on Finite State Machine (FSM) model. This FSM model will help them to deduce conclusions. This approach is capable of detecting Cross Site Scripting (XSS) based attacks. But this approach fails when there are embedded objects in the page. In our approach, we also consider the hyperlinks and their source not just the content of the webpage.

Alkhateeb *et al.* [1] developed a RDF based phishing detection method; it inspects legitimacy of the suspicious webpage by extracting RDF features from a bank's web page and checks it with predefined RDF knowledge base maintained at the centralized server. In order to accomplish this, each bank's profile has to be maintained at the centralized server. Profile information contains details like the bank's URL, bank name, branch name, allowed ports, allowed Internet Protocol (IP) addresses etc., This basic information about the bank is constructed as an ontology in the database for future comparisons.

But, the drawback of this approach is, it detects phishing websites that aims at only banks. Also, this method possesses practical issues.

## 3. Proposed Architecture

Our method for identifying phishing pages consists of two stages. First stage, explores the strength of RDF in identifying phishing web pages based on their metadata. Second stage, explores a machine learning technique that can help our system in taking a decision given a feature set as input. We will discuss why we have adopted two stages and also justify the need for both the stages and their significance in detail in coming sections.

The entire system is viewed as three step process as shown in Figure 2.

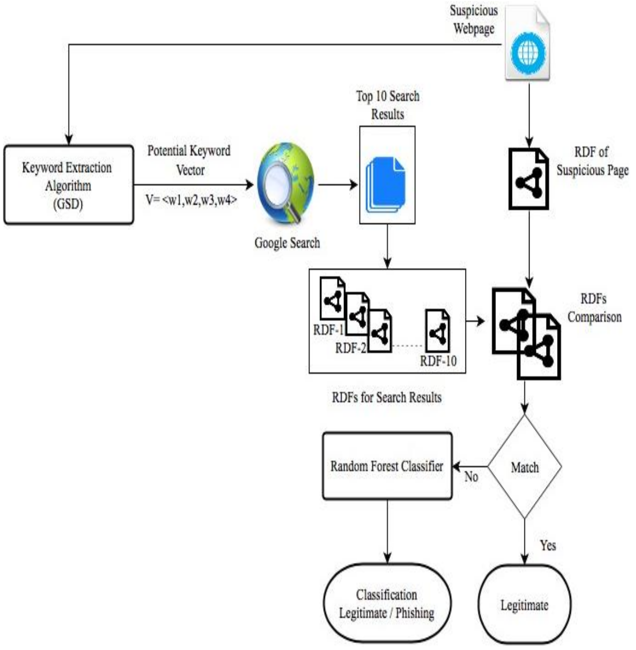

Figure 2. System architecture.

First, given a suspicious webpage, our system extracts the metadata from the source code and content of the page and then constructs an RDF structure for the page. Second, keywords are extracted from the given suspicious page and this keyword vector is fed to a search engine. Top "n" results are collected in the same order as appeared. RDFs are created for all the web pages that has come as top results in the search. Third, a comparison is made between the RDF of suspicious page and RDFs of the search results to come to a conclusion. If this previous stage fails without giving a conclusion, then we employ the second stage, where a feature vector is given as an input to Random forest classifier to take a decision.

### 3.1. RDF in Decision making

#### 3.1.1. RDF Construction

Hypertext Markup Language (HTML) to RDF model generation is done after extracting set of features (metadata) from the suspicious web page. These features are our element set that we use for constructing RDF model for the page. We have chosen 21 features that can be used to differentiate between phishing and legitimate pages. These features are chosen in such a way that no two different web pages will have the same element set. There are already some defined vocabularies for RDF like Dublin Core [6], Extensible Hypertext Markup Language (XHTML) [19] and HTTP [8] vocabularies. We have also added our own elements to the existing element set and also evaluated the strength of all the elements based on a comparative study on different web sites from various domains. Some properties from the above mentioned vocabularies are not considered as observations found that they are not suitable for the current problem that we are addressing. The element set that we have chosen for this approach is given in the Table 1.

Table 1. Element set.

| S.No | Properties | Description | Source |
|---|---|---|---|
| 1 | Title | Name or Title of the given web page. | Dublin Core |
| 2 | Parent Domain Name | Give the character set format of the web page. | Novel |
| 3 | Creator | An entity primarily responsible in creation of the web page. Examples of a Creator include a person or an organization. | Dublin Core |
| 4 | Subject | Typically, the subject will be represented using keywords, key phrases of the web page. | Dublin Core |
| 5 | Description | A brief description of the web page. | Dublin Core |
| 6 | Creation Date | Gives the creation date as well as the last modified date of the web page. Can be obtained from WhoIs lookup | Dublin Core |
| 7 | Identifier | URI of the web page. | Dublin Core |
| 8 | Status code | Gives the status code number that is returned from HTTP response. | HTTP |
| 9 | Form Action | Action tag of the form, checking this avoids XSS attacks | Novel |
| 10 | Form name | Name given to the form element | XHTML |
| 11 | Form method | GET or POST method of the form | Novel |
| 12 | Imgsrc | Gets the Source of the images | XHTML |
| 13 | Age of Domain | Gets the age of the domain by doing WhoIs lookup | Novel |
| 14 | Copyright | Gives the copyright information of the web page. | Novel |
| 15 | Captcha | Check for captcha in the web page | Novel |
| 16 | Frame src | Gets the source of the frames in web page | Novel |
| 17 | IP address | Gets the domains IP addresses of the web page | Novel |
| 18 | Updated date | Last Updated date of the Domain | Novel |
| 19 | Expiration Date | Expiration date of the Domain | Novel |
| 20 | Registrar | Registrar under which the Domain is registered | Novel |

All the above mentioned features are represented as RDF properties each represented using a triplet (subject, predicate and object). These features extracted from the web page are represented as RDF statements forming RDF model for the page. Sometimes suspicious web page or even legitimate web pages may not contain any features except frames; in that case content is extracted from the source of the Frame. Each and every statement is represented in the form of a triple making the logical assertions simple. We have designed our own RDF schema that is used to create the RDF models for the web pages. RDF schema defines the class, subclass, property, sub property relations between the elements giving way for logical assertions.

### 3.1.2. Keyword Extraction

Keyword extraction plays a vital role in our approach. Extracting potential keywords from the suspicious web page will give us the best possible results in the search when these keywords are given to a search engine. In order to achieve this, we employ a streamlined process where we extract the keywords from the meta tags ‘title’ and ‘keywords’ if they are present in the source code of the page. Some pages may contain only frames without any body, in that case we extract the body content from that frame “*src*” tag. We extract the body content from the web page and perform some basic text processing steps like stop word removal, stemming etc.,

Once pre-processing is done, we perform keyword extraction based on the Google Similarity Distance Algorithm [5]. This algorithm uses different metrics like Normalized Information Distance (NID), Normalized Compression Distance (NCD) and Normalized Google Distance (NGD) metrics to extract the potential keyword vector from the web pages. This algorithm is a well tested and highly appreciated algorithm. Specially, this algorithm works better on web sites than any other keyword extraction algorithm.

We employed this keyword extraction algorithm in our approach as it best suits our problem. This algorithm calculates the relationship between two words using NGD score where NGD between two words is calculated using the formulae:

$$NGD(x,y) = \frac{G(x,y) - min\left(G(x),G(y)\right)}{max\left(G(x),G(y)\right)} = \frac{max\left\{log\ f(x), log\ f(y)\right\} - log\ f(x,y)}{log\ N - min\left\{log\ f(x), log\ f(y)\right\}} \tag{1}$$

Where $f(x)$ and $f(y)$ are number of search results of the word ‘x’ and ‘y’ respectively. $f(x,y)$ is the number of web pages which contain both ‘*x*’ and ‘*y*’. Thus, Google similarity distance can be used to calculate the relationship between every two words. Finally these words with close relationship are arranged in a proper order to form potential keyword vector that can be given as an input to a search engine like Google to get the best matching results.

### 3.1.3. Phishing Detection

After constructing the RDF model for the suspicious webpage and RDF models for the web pages obtained from the search results, a comparison is done between RDF model of suspicious page and each of the RDF model obtained from the search results. We compare the RDFs using graph isomorphism. Two RDF models are said to be isomorphic, if each statement in one RDF can be matched with a statement in other RDF.

Comparing RDFs is mentioned clearly in a paper by Hewlett Packard (HP) [3]. This research carried out by HP reveals how the standard graph isomorphism algorithms can be used for comparing two RDFs. Therefore, when a RDF model of a suspicious web page is matched isomorphic with any of the RDFs model obtained from the search results, then that suspicious page is considered to be legitimate. As we know search engines like Google are based on the concept of page ranking, there are less or almost zero chances for a phishing web page to come up in the top

10 results of the search, this gives us very negligible or zero false negatives in the proposed system. Therefore, we can say that if isomorphic match is success, the suspicious page is legitimate. This stage consumes very less time to predict the nature of the suspicious page. If the suspicious page is a legitimate one, our system can predict it within few seconds. This is one major advantage of this approach where, the system will not take much time to predict a legitimate website as legitimate.

A website is a collection of web pages. For example a website like Flipkart will have ‘n’ number of pages including home page, login page, register page etc., there can be some web pages that are legitimate but may not appear in the top results of search because of a poor keyword vector. Sometimes the exact page we are looking for may not be in the top results, but definitely the results will be from the same website domain from which the page is hosted.

If the keywords extracted are potential and strong, then the system performance is highly appreciable. If the keywords extracted are not potential and weak, then the chances of the exact page coming up in the top results cannot be predicted, this situation may give room for false positives in the system as we check for RDF isomorphism. RDF isomorphic function will return false even if a single element does not match within the element set. False positives in the system will affect the accuracy of the system. Therefore, in order to reduce the number of false positives in the system we employ the stage-2. Stage-2 is invoked only when the isomorphic case fails in the first stage.

When the isomorphic case fails, our system will not directly consider the suspicious page as phishing, instead the system generates a feature vector from the element set and it is given as input to Random forest classifier to classify the suspicious page appropriately. This gives the system much better performance as both stage one and two go hand in hand to improve the accuracy of the system.

### 3.1.4. Decision Making Using Random Forest Classifier

Random Forests is an Ensemble approach for classification and regression. Random Forest classifier constructs number of decision trees during the training time and outputs a class that is the mode of the classification classes of the individual trees. Random Forest classification performs better than any other decision tree algorithms as it uses a forest of classification trees to take a decision [10]. In Random Forests, to classify a new object from an input vector, we give the input vector each of the trees in the forest.

Each tree gives a classification, and we say that the tree ‘votes’ for that class. The forest chooses that classification which has more votes over all the trees in the forest. The training algorithm for Random forests applies the general technique called Bagging. Given a training set of N size, X=$x_1$,$x_2$,.....$x_n$ with responses Y=$y_1$,$y_2$,.....$y_n$, Bagging selects a random sample from the training set with replacement and try fitting trees to these samples. If there are V variables in the input vector, Random forest uses a modified tree learning algorithm that selects a random subset of features at each candidate split in the learning process. This random selection of features sometimes referred as “feature bagging”. Typically if a dataset is having R features, √V features are used at each split.

We have done a comparative study between Support Vector Machines (SVM), Decision tree algorithms like C4.5 and Random forests. One of the disadvantage with SVMs is that they can be sorely inefficient to train. So, it is not recommended for our problem which have many training examples. ANN becomes complex when there are more number of hidden layers. With Decision Tree classifiers, the problem is over fitting; they do not generalize well with the training data and have low prediction accuracy and highly biased decisions. So we have considered the Tree ensembles. The two important algorithms in Tree ensemble are Random Forests and Gradient Boosted Trees (GBT). Out of which the GBT are harder to be tuned and are prone to over fitting.

Random Forest have good accuracy even with missing values, and are less sensitive to outliers and parameter choices. In RDF construction phase, there is a chance that we may not be able to get data for all the element set from a web page, i.e., there is a scope for missing values in our approach. As random forests handle the missing values appropriately, we adopted it in our approach so as to improve the overall accuracy of the system.

Our training dataset consists of 1126 Phishing and 952 legitimate samples collected from different sites as mentioned in Tables 2 and 3 shown below.

Table 2. Legitimate data source.

| Source | Sites | Link |
|---|---|---|
| Google’s top 1000 most-visited sites | 640 | http://www.google.com/adplanner/static/top1000/ |
| Alexa’s Top sites | 160 | http://www.alexa.com/topsites |

Table 3. Phishing data source.

| Source | Sites | Link |
|---|---|---|
| PhishTank’s open database | 944 | http://www.phishtank.com/ |
| Reasonable-Phishing Web pages List | 312 | http://antiphishing.reasonables.com/BlackList.aspx |

Our feature vector consists of 12 different features. Features are selected in such a way that these features apply to a complete domain but not for individual pages. Our feature vector is given as follows:

- V=<Identifier, Age of domain, Copyright, Presence of Captcha, IP addresses, Parent domain name, creation

date, Last updated date, expiration date, status code, Registrar Name, Name Servers>

In order to understand the importance of this feature vector and random forests, please consider an example where input is a phishing URL of PayPal. Keywords are extracted from the phishing site and given to a Google search engine. If top 10 results from Google has the exact page that we are looking for, then the isomorphic match will be a success and suspicious URL is considered as legitimate. Let's say, in case the top 10 results from the search does not contain the exact page that we are expecting. In this case, isomorphic comparison will fail with all the 10 results.

At this stage we cannot just say that the suspicious URL is a phishing one. This can happen if the keyword vector couldn't bring the best possible results from the search. To reduce this kind of false positives, we go for Random forest classifier, where the feature vector is dependent on domain but not on a single page from the domain. A PayPal website can contain many pages but features like domain name, IP address, creation date, Name servers etc. all will be same throughout the PayPal domain and they don't change from page to page in PayPal. This gives us strength to go for a further step where random forest classifier can take a decision based on this feature vector. This also gives our approach a chance to predict the possible phishing target, in this case it is Paypal.com.

## 4. Implementation

Our system implementation is majorly done in Java language. Apache Jena [2] is a free open source Java framework for building semantic web applications.

We have used this framework for building the RDF schema and validating it. Web scraping is done using Jericho HTML parser [9]. It is a Java library having built-in functionality to extract all text from HTML mark-up and best suitable for feeding into a text search engine. The extracted text from the suspicious web page will be an input to the keyword extraction algorithm. Text processing is done by applying tokenization, stop word removal using Stanford's stop word list and stemming. Processed text is given to Google Similarity Distance algorithm to obtain the potential keywords. For training our dataset, we have used Rapid Miner machine learning tool [15] to implement Random forest classifier and performed 10-fold cross-validation in which the dataset is divided into 10 parts. Out of the 10 sub samples 9 parts will be used for training and validation is done on the 10th part. We chose Rapid Miner as it can be easily integrated with Java based applications. Dataset of phishing web sites has been obtained from PhishTank, which is a reliable source as all the sites are reported based on peer reviews.

## 5. Results

We have used three metrics to evaluate the performance of system, which are True Positive Rate (TPR), False Positive Rate (FPR) and Accuracy (ACC).

True Positive Rate (TPR) measures the percentage of correctly classified phishing sites. TPR is computed using Equation.

$$TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)} \tag{2}$$

True Positives (TP) is the number of correctly classified phishing pages. P is the number of phishing pages, which is equivalent to sum of correctly classified phishes TP and falsely classified phishes which is False Negatives (FN).

FPR measures the percentage of legitimate sites wrongly classified as phishing. FPR is computed using Equation.

$$FPR = \frac{FP}{L} = \frac{FP}{(FP + TN)} \tag{3}$$

Here False Positives (FP) is the number of legitimate pages which are wrongly classified as phishing, L is the number of legitimate pages which is equivalent to sum of falsely classified legitimate pages FP and correctly classified legitimate pages which is True Negatives (TN).

Accuracy (ACC) measures the degree of closeness between measurements of classified sites and sum of actual phishing sites and legitimate sites. ACC is computed using Equation.

$$ACC = \frac{(TP + TN)}{(P + L)} \tag{4}$$

Here accuracy value will be close to 100 for any ideal anti phishing system. Accuracy of the system can be improved by having higher TP value and lower FP value.

- *Data Sources.* The different data sources that we used to collect the dataset and the respective evaluation results are given below:

Table 4. Experiment results: N is the total number of pages, n is the number of correctly classified pages.

| | Phishing pages | Legitimate Pages | Total |
|---|---|---|---|
| N | 1256 | 800 | 2056 |
| n | 1241 | 788 | 2029 |

- *Results.* The experiment results are shown in Table 4. The true positive rate of this method is 98.8%, false positive rate is 1.5% and accuracy is 98.68% as shown Figure 3. This statistics clearly shows that this system detects phishes with less false positives and high accuracy rate. Moreover, for all the successfully classified pages we have identified its target also.

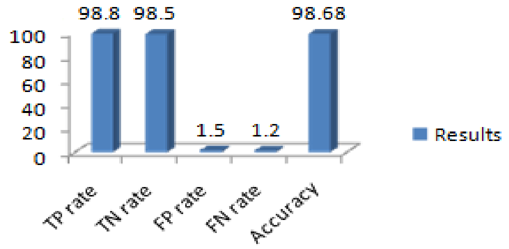

Figure 3. Results of proposed system.

## 6. Conclusions and Future Scope

In this paper, we have come up with a new method for identifying phishing web sites. Our goal is not just identifying phishing web sites, but also to provide with the possible targeted domain. We employed a two stage process where the first stage is based on RDF model of the web pages and the second stage is based on a machine learning technique. Both stages work hand in hand to reduce the number of false positives and to improve the system's accuracy. As we have employed a better keyword extraction algorithm, our system has very less, almost zero false negatives.

Our future work, includes converting this RDF models to ontologies and making use of (Web Ontology Language) OWL with ensemble approaches to predict phishing attacks.

## References

[1] Alkhateeb F., Manasrah A., and Bsoul A., "Bank Web Sites Phishing Detection and Notification System Based on Semantic Web Technologies," *International Journal of Security and its Applications*, vol. 6, no. 4, pp. 53-66, 2012.

[2] Apache Jena: A free and open source Java framework for building semantic web and linked data applications, Available at https://jena.apache.org, Last Visited, 2015.

[3] Carroll J., "Matching rdf Graphs," HP Laboratories Technical Report HPL 293 (2001).

[4] Chou N., Ledesma R., Teraguchi Y., and Mitchell J., "Client-Side Defense Against Web-Based Identity Theft," *in Proceedings of the 11th Annual Network and Distributed System Security Symposium*, San Diego, pp. 1-16, 2004.

[5] Cilibrasi R. and Vitanyi P., "The Google Similarity Distance. Knowledge and Data Engineering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370-383, 2007.

[6] Dublin core metadata initiative, Available at http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=elements#, Last Visited, 2015.

[7] Fette I., Sadeh N., and Tomasic A., "Learning to Detect Phishing Emails," *in Proceedings of the 16th International Conference on World Wide Web*, Banff, pp. 649-656, 2007.

[8] HTTP vocabulary, Available at http://www.w3.org/2011/http#, Last Visited, 2015.

[9] Jericho HTML Parser, Available at http://jericho.htmlparser.net, Last Visited, 2015.

[10] Kremic E. and Subasi A., "Performance of Random Forest and SVM in Face Recognition," *The International Arab Journal of Information Technology*, vol. 13, no. 2, pp. 287-293, 2015.

[11] Muppavarapu V., Gowtham R., and Archanaa R., "An RDF based Anti-Phishing Framework," *International Journal of Software and Web Sciences*, vol. 1, no. 9, pp. 1-10, 2014.

[12] Pan Y. and Ding X., Anomaly Based Web "Phishing Page Detection," *in Proceedings of 22nd Annual Computer Security Applications Conference*, Miami Beach, pp. 381-392, 2006.

[13] PhishTank Phishing Database, Available at http://www.phishtank.com/, Last Visited, 2015.

[14] Prakash P., Kumar M., Kompella R., and Gupta M., "Phishnet: Predictive Blacklisting to Detect Phishing Attacks," *in Proceedings IEEE INFOCOM*, San Diego, pp. 1-5, 2010.

[15] Rapid Miner Data Mining and Machine Learning Tool, Available at https://rapidminer.com/, Last Visited, 2015.

[16] RSA Anti-Fraud Command Center, RSA Monthly Online Fraud Report, January 2015: http://www.emc.com/collateral/fraudreport/h13929-rsa-fraud-report-jan-2015.pdf, Last Visited, 2015.

[17] RSA Anti-Fraud Command Center, RSA Monthly Online Fraud Report, Available at http://www.emc.com/collateral/fraud-report/online-fraud-report-1114.pdf, Last Visited, 2015.

[18] Shahriar H. and Zulkernine M., "Trustworthiness Testing of Phishing Websites: A Behavior Model-Based Approach," *Future Generation Computer Systems*, vol. 28, no. 8, pp. 1258-1271, 2012.

[19] XHTML vocabulary, version date: 2010-01-27, http://www.w3.org/1999/xhtml/vocab#, Last Visited, 2015.

[20] Xiang G., Hong J., Rose C., and Cranor L., "CANTINA+: A Feature-Rich Machine Learning Framework For Detecting Phishing Web Sites," *ACM Transactions on Information and System Security*, vol. 14, no. 2, pp. 1-32, 2011.

[21] Zhang J., Porras P., and Ullrich J., "Highly Predictive Blacklisting," *in Proceedings of the 17th Conference on Security symposium*, San Jose, pp. 107-122, 2008.

[22] Zhang Y., Hong J., and Cranor L., "Cantina: A Content-Based Approach To Detecting Phishing Web Sites," *in Proceedings of the 16th*

*International Conference on World Wide Web*, Banff, pp. 639-648, 2007.

**Vamsee Muppavarapu** is an Assistant Professor in the Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham University. His primary research interests include anti-phishing, semantic web and recommender systems. He received his Master's degree in Computer Science from Amrita Vishwa Vidyapeetham University in Coimbatore, India.

**Archanaa Rajendran** is an Assistant Professor in the Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham University. Her research interests include machine learning, recommender systems and data mining. She received her Master's degree in Computer Science from Amrita Vishwa Vidyapeetham University in Coimbatore, India.

**Shriram Vasudevan** is an Embedded System Engineer with about ten yearsof experience in the IT and academics. He has authored 28 books for variousreputed publishers across the globe. He has also written a lot of researcharticles. He has been awarded by Intel, IEI (India), Wipro, Infosys, ICTACT, CII, Computer Society of India, and VIT University, etc. for his technical contributions. He received his Mastersand Doctorate in Embedded Systems. He is currently associated with Amrita Vishwa Vidyapeetham, India. He was associated with WiproTechnologies, Aricent Technologies and VIT University.