WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Anti-Phishing Technology Using Neuro-Fuzzy Approach On Fog Networks (Fog Phishing)

Achu Thomas Philip[1], Bibin Varghese[2], Dr. Smita C Thomas[3]

[1] M Tech Student, APJ Abdul Kalam Technological University, Kerala, India
[2] Asst. professor, Mount Zion College of Engineering, Kadammanitta, Kerala, India
[3] Professor, Mount Zion College of Engineering, Kadammanitta, Kerala, India

ABSTRACT

Phishing is a technique to gain personal information for the purpose of identity theft, usually by means of fraudulent E-mail. Attackers use emails, social media to trick victims into providing sensitive information or visiting malicious URL (Uniform Resource Locator) in the attempt to compromise their systems. In most servers contains websites which has phishing behaviors. They stoles the sensitive data from the user and that causes the user an economic loss or data loss. The current anti-phishing techniques are difficult to apply real time detection of phishing sites. So there is a secured and transparent system needed which enable the user to validate the site directly. This system mainly focused on performing a real time detection of a phishing site. The main objective of this is for better identification of sites in the category legitimate or phishing. Based on a built Neuro-fuzzy framework, we propose using uniform resource locator (URL) features and online traffic data to detect phishing websites in this paper (dubbed Fi-NFN). We develop an anti-phishing model based on Cisco's innovative fog computing method to transparently monitor and defend fog users from phishing assaults. Our suggested approach's experiment findings, which were based on a large-scale dataset collected from real phishing cases, showed that our system can effectively prevent phishing attempts and improve network security.

Key Words: Phishing, Anti-phishing, Fog Computing, Neuro-fuzzy system.

1. INTRODUCTION

Phishing is a type of social engineering attack that seeks to take advantage of a flaw in the system at the user's end. For example, a system may be technically secure enough to prevent password theft, but an unwitting user's password may be leaked if the attacker sends a falsified (phished) update password request. Phishing is analogous to fishing in the water, except that instead of catching a fish, attackers attempt to steal personal information from consumers. When a person visits a bogus website and inputs their login and password, the attacker obtains the victim's credentials, which can be used for harmful reasons.

The term "phishing" comes from the word "fishing." Because phishing websites look so much like legitimate ones, online consumers can easily be duped into providing personal information. Phishers utilise a variety of strategies to deceive their victims when they create phishing sites, including email messages, instant chats, forum postings, phone calls, and social networking information. Phishing causes significant economic damage all around the world, and the number and complexity of phishing sites is continually increasing. The number of phishing attacks is increasing by 5% monthly, according to reports from the Anti-Phishing Working Group.

However, the anti-phishing problem has not been well addressed at the network edge for the following reasons. Many research employ the blacklist/whitelist strategy. However, all methods require a manual/automated update procedure to maintain a list of phishing websites. Before launching, URL requests are validated against a local database or a cloud database. Despite the fact that blacklist/whitelist approaches detect phishing sites quickly, administering the blacklist/whitelist database for both the local and cloud databases is inefficient due to the continually expanding number of phishing sites. However, the anti-phishing problem has not been well addressed at the network edge for the following reasons. For starters, mobile users are more likely than desktop users to check their emails and utilise online browsers.

As a result, people are far more likely to visit phishing sites that have not yet been discovered or blocked by anti-phishing software or firewalls on their local networks or devices. Second, because mobile devices are continually "hungry" for energy and computational resources, anti-phishing software is frequently neglected or disabled. As a result, it's difficult for consumers to tell if an incoming link is real or not. Third, existing anti-phishing solutions are ineffective at detecting phishing attempts, and mobile users may be vulnerable to phishing attacks while performing routine. tasks According to the survey, mobile users are three times more likely than desktop users to submit their login credentials. As a result, phishing attempts against terminal users are a serious concern at the network's edge.

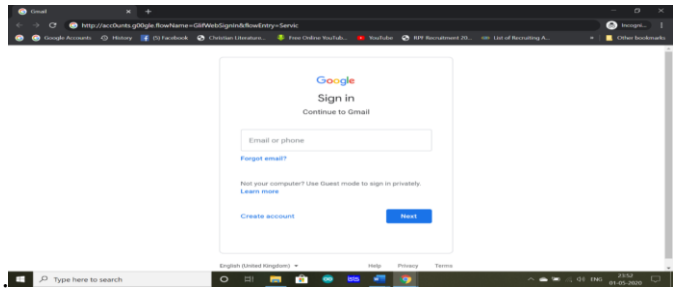


Fig. 1: Gmail Phishing Scam URL

The Fig. 1 sign-in form looks precisely like a Gmail sign-in form, with the exception that the URL has been slightly modified. However, filling out this form would not grant the attacker complete access to the victim's Gmail account. The type of theft and fraud that may occur simply by obtaining the data of someone's or some organization's account is unimaginable. The Gmail account is in charge of all other accounts. That might pose a serious hazard. Microsoft Outlook fraud is the second most common, followed by Google Drive. Facebook, bank logins, Pay-tm, Pay-pal, and other sites are also targets.

The following is an example of a phishing attack. The following is an example of a common phishing scam: A hoax email purporting to be from myuniversity.edu is sent to as many professors as possible. The user's password is due to expire, according to the email. Within 24 hours, they must go to myuniversity.edu/renewal and renew their password. When you click the link, a variety of things can happen. For instance, the user is sent to myuniversity.edurenewal.com, a false page that looks identical to the real renewal page and asks for both new and existing passwords.

1.1 ANTI-PHISHING

Anti-phishing is a technology service that aids in the prevention of unwanted access to secure and/or sensitive data.

1.1.1 ANTI-PHISHING SERVICES

An anti-phishing service combats a specific type of effort to obtain personal or sensitive data. While anti-phishing services give tools to assist users in recognising Web phishing, many anti-phishing features are responses to attempts to breach a system and steal data. Many phishing attempts are made using web apps, hence certain anti-phishing techniques are offered through web applications. Some anti-phishing services offer advanced planning to assist clients in avoiding data theft. For example, a good "phishing incident response strategy" requires a timely response to illegal access. Anti-phishing services or technologies frequently include specialised components that assist in determining how data is taken, how data may be restored, or how to close ranks and defend a system from further hacking. Experts predict that more complex phishing will emerge, thus new anti-phishing services will often address this directly with more inventive features and components.

1.2 FOG COMPUTING

Fog computing, also known as fog networking or fogging, is a decentralised computer architecture that sits between the cloud and data-generating devices. Users can deploy resources, like as programmes and the data they produce, in logical areas to improve performance using this flexible framework.

1.2 NEURO-FUZZY SYSTEM

A Neuro-fuzzy system is a fuzzy system that determines its parameters (fuzzy sets and fuzzy rules) by processing data samples using a learning method developed from or influenced by neural network theory.

2. LITERATURE SURVEY

2.1 Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach

A novel framework for content-based phishing web page detection is given, which employs a Bayesian method. To quantify the resemblance between the protected web page and suspect web pages, our approach considers both textual and visual content. We introduce a text classifier, an image classifier, and an algorithm that combines the outputs of classifiers. The use of a Bayesian model to determine the matching threshold is a standout element of this paper.

This is necessary by the classifier in order to determine the web page's class and determine if it is phishing or not. The naive Bayes rule is employed in the text classifier to calculate the chance that a web page is phishing. The earth mover's distance is used to quantify visual similarity in the image classifier, and our Bayesian model is used to estimate the threshold. The Bayes theory is employed in the data fusion technique to synthesis the classification results from textual and visual content.

The effectiveness of our proposed approach was examined in a large-scale dataset collected from real phishing cases. Experimental results demonstrated that the text classifier and the image classifier we designed deliver promising results, the fusion algorithm outperforms either of the individual classifiers, and our model can be adapted to different phishing cases.

2.2 Gold Phish: Using Images for Content-Based Phishing Analysis

Human-vulnerability-based attacks have become more common in recent years. Phishing assaults are one of the most popular types of attacks in this category. Phishing attempts typically employ emails to persuade unwary users to enter sensitive information such as credit card numbers or bank account details onto bogus websites. The bogus websites are created to look just like the real ones.

The Uniform Resource Locator (URL) is frequently identical to the actual site's URL. Even the most computer-savvy people have been known to fall prey to phishing sites, according to studies. There were 30,131 unique domain names undertaking phishing attacks in the first half of 2009. Phishing assaults were blamed in late 2007 for a loss of more than $3 billion, according to a research. According to the same research, phishing scams cost 3.6 million people money. Many tools have been created to counteract phishing assaults. The majority of anti-phishing techniques in use today rely on databases that generate a blacklist of known phishing sites. There are a lot of drawbacks to this strategy. For starters, this method is based on a comprehensive database of all known phishing sites. In this regard, the anti-phishing programme is only as good as its database's completeness. This is exacerbated by the fact that the average phishing site is only live for a few days, and some are only active for a few hours. Second, this strategy does not guard against zero-day phishing attacks, which are brand-new phishing attempts that the general public is unaware of. Every day, on average, 82 new phishing sites appear. We present a method for identifying zero-day phishing assaults, which eliminates the limitations that database approaches have.

Our solution, Gold-Phish, detects and reports phishing sites using a browser plug-in. We achieve this by reading the text from an image of the website (particularly, the corporate logo), obtaining the top ranked domains from a search engine, and comparing them to the current web site using optical character recognition (OCR). The user's ability to recognise well-known corporate logos is the tool's strength. A phishing site cannot hide the changing of a well-known company logo from the phishing victim.

2.3 Real Time Detection of Phishing Websites

Web spoofing entices users to connect with bogus websites instead of the actual ones. The primary goal of this assault is to steal confidential information from users. The attacker develops a'shadow' website that appears to be identical to the original site. This deceptive conduct gives the attacker access to the user's information and allows them to modify it. This study presents a phishing website detection technique based on inspecting web page Uniform Resource Locators (URLs). By verifying the Uniform Resources Locators (URLs) of suspected online pages, the suggested approach is able to discern between authentic and false web pages.

To check for phishing web pages, URLs are examined based on specific features. Phishing websites imitate the appearance of authentic websites in order to lure a large number of Internet users. The discovered attacks are reported in order to avoid future attacks. The suggested solution's performance is assessed using the Phis-tank and Yahoo directory datasets. The acquired findings demonstrate that the detection method is deployable and capable of detecting various forms of phishing attempts while minimising false alarms.

2.4 Phishing Websites Classification using Association Classification (PWCAC)

Phishing is one type of cybercrime that appears to be on the rise, with unwary users being targeted. Phishing's reach has now expanded to include companies who provide services over the Internet. Because of phishing, such businesses are more likely to lose their reputation and competitive advantages. The status of a set of significant qualities that exist in the website is used to classify a website as phishing or legitimate. To combat phishing assaults, a number of solutions have been created.

There is, however, no solution that has been able to totally solve the problem. The proposed association classification method is applied to the well-known phishing websites dataset in the UCI repository in this research. In terms of many evaluation criteria that are typically used in evaluating classification data mining domains, the trial results were promising.

3. EXISTING SYSTEM

Many studies have been conducted in the topic of phishing detection in the past. We obtained data from a variety of sources and thoroughly examined it, which aided us in driving our own approaches in the development of a more safe and accurate system.

Blacklist Approach and Whitelist Approach

Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta (2010) developed a predictive blacklist strategy for detecting phishing websites in their paper. It used heuristics and an adequate matching technique to find new phishing URLs. Heuristics combined pieces of known phished websites from the available blacklist to build new URLs. The URL's score is then calculated using the matching algorithm. This website is flagged as phishing if the score exceeds a certain threshold number.

Heuristic Approach

Aaron Blum, Brad Wardman, and Thamar Solorio offered a paper in which they explored surface level data from URLs in order to train a confidence weighted learning algorithm. The objective is to limit the source of available characteristics to the URL's character string in order to eliminate the risk of extracting host-based data. Every URL is represented as a binary feature vector[4]. These vectors are fed into an online algorithm, which maps previously unseen URLs in the binary feature vector to it at the time of testing.

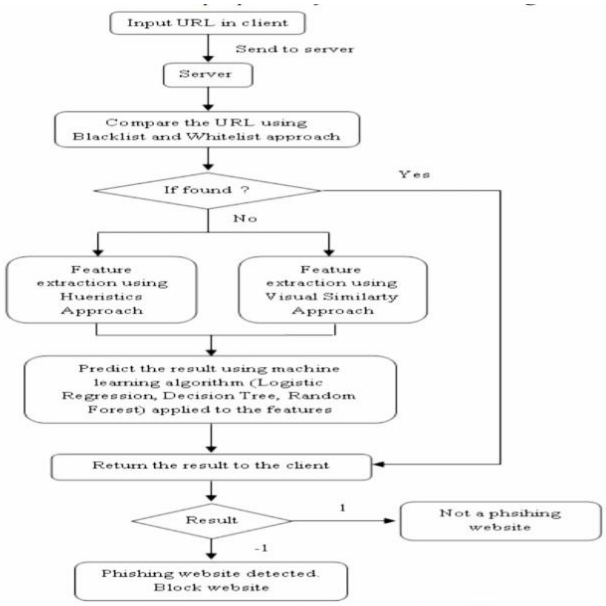


Fig 2: Flowchart of Existing System

Visual Similarity Approach

A hybrid method based on URL and CSS matching was given by A. Mishra and B. B. Gupta in. This method can detect embedded noisy contents, such as a picture on a web page, which is utilised to maintain the webpage's visual similarity. They used Jian Mao's, Pei Li's, Kun Li's, Tao Wei's, and Zhenkai Liang's CSS similarity comparison technique in their technique. Text content and text features are two separate forms of visual features. Font colour, font size, backdrop colour, font family, and so on are all text features. Because the attacker takes the page content from the original website, this strategy fits the visual aspects of other websites.

4. PROPOSED SYSTEM

Every day, a new website is launched. Because there is a vast amount of data relating to the sites available on the internet, data mining is an interesting issue. Because predicting phishing sites is a difficult task, we used Kaggle to conduct our research. We constructed a dataset, processed it, and employed machine learning techniques to create efficient models that can predict phishing sites. The receiver operating characteristic curve, commonly known as the ROC curve, shows genuine positives vs false positives.

4.1 METHODOLOGY

Phishing detection is considered a criminal issue in the field of Internet security. These current hardware-based technologies provide an additional layer of defence against phishing assaults by putting a gateway anti-phishing in the networks. Due of the variety of phishing assaults, such hardware devices are expensive and inefficient to operate. An anti-phishing gateway can be built as software at the network's edge, with embedded powerful machine learning techniques for phishing detection, thanks to promising virtualization technologies in fog networks.

Based on a built Adaptive Neuro-Fuzzy Inference System, we leverage unified resource locator (URL) properties and online traffic features to detect phishing websites in this research (ANFIS). Based on the novel method, fog computing, as promoted by Cisco, creates an anti-phishing model that monitors and protects fog users from phishing assaults in real time. To identify phishing sites, the author use well-known ranking methods. They appear to be identical; yet, combining them can improve detection accuracy for the following reason. First, the Google Index system returns empty values for newly established URLs, but others can compensate with positive values.

Second, while Google Index isn't a ranking system, it does have a large dataset and reliable findings. This combination accurately reflects the lifespan of URLs. Other characteristics, such as special characters in URLs or the number of dots or the length of the URL, can be used to identify phishing sites, but they are quite particular, and attackers can quickly modify or fake them. The goal of this research is to detect phishing assaults in real time. As a result, the system has less time to analyse and decide. As a result, avoid choosing identification features that can't be analysed in real time. The identification component is built within a fog node and communicates with fog users[1].

It includes an ANFIS network that has already been trained to categorise URLs into two groups: phishing URLs and authentic URLs. There is a link between these components that allows the ANFIS network's training parameters to be updated. In comparison to the blacklist, this procedure does not need a lot of network traffic or time to update the phishing database. Moreover, administrators can quickly launch and alter the training procedure in the back-end component. Finally, in a fog node, the training and updating phases have no bearing on the identification phase.

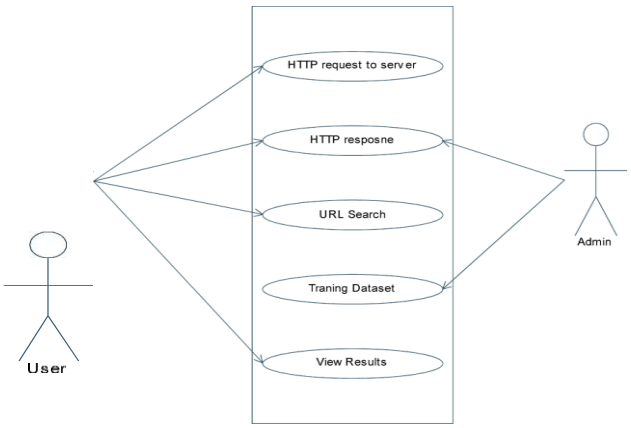
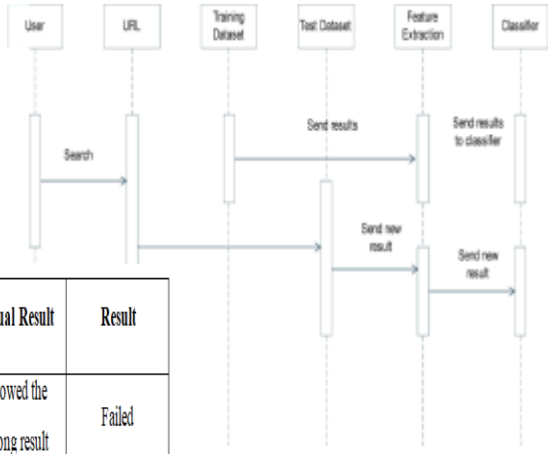


Fig 3. Use Case Diagram

We initially enter our source code into the use case diagram, after which we select useful code for testing. Then verify the promise data repository to see if the software's problematic dataset is present. If a software-related dataset is discovered to be defective, read the data set in csv format. Then, using 0 and 1, define the class names such as defective and non-defective. Then split the dataset into two parts: test data and training data.

Then, by leveraging historical data of software components and their flaws, we apply Machine Learning approaches to construct detections regarding software component failure. In terms of detecting software defects, machine learning approaches are important. Defective and non-defective software components are separated. For the classification of defective data sets, Decision Tree and Logistic Regression techniques are used. The datasets are obtained from the promise data repository, and accuracy is calculated as a result. The algorithms are implemented with the Weka tool and Python language, with comparative analysis of the outcomes displayed.



ig 4. Sequence diagram

Module Name	Test Case	Expected Result	Actual Result	Result
User Module	To enter a Site	Must Show the Result	Showed the wrong result	Failed
User Module	To enter a Site	Must Show the Result	Showed the wrong result	Success

Table 1 Test Case

Module Name	Test Case	Expected Result	Actual Result	Result
User Module	To enter a Site	Must Show the Result	Showed the wrong result	Failed
User Module	To enter a Site	Must Show the Result	Showed the wrong result	Success

5 MODULE DISCRIPTION

There are mainly 2 modules

Mining Module

Mining module is about the processes in the system for learning and predictions. These are the background functions of the system. The developer is responsible for this.

- Data Extraction
- Data Pre-processing
- Data Integration and Transformation
- Feature Selection
- Classification

Web Module

This module is where user is involved. The user can give actual inputs and gets output.

- Actual Data Input
- Output

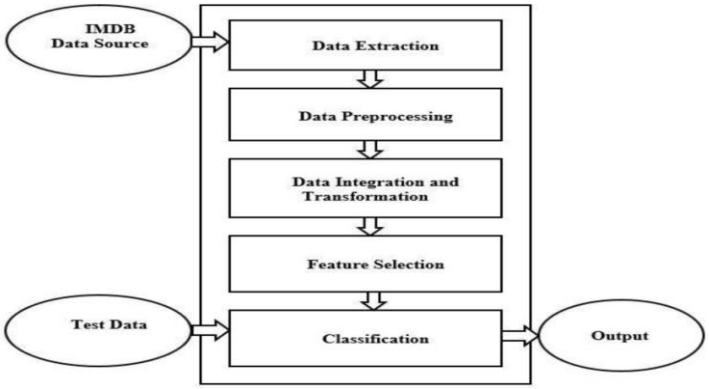


Fig.5 Architecture

Data Extraction

The dataset to be used is collected from kaggle.com

1) Data Pre-processing

The data we extracted from kaggle need to be cleaned as the data is obtained from multiple sources.

2) Data Integration and Transformation

The data retrieved from kaggle must be combined and modified before being used for analysis and classification. We must generalise the rate for analysis and classification because the values will be continuous. This will be our prediction class.

3) Feature Selection

The process of feature selection and identification can be done in a variety of ways. The issue of the most important attribute from the dataset is addressed by feature selection. It gets rid of the attributes that don't bring any value to the analysis. It determines the weight of the most contributed attribute and the least contributed attribute.. There are many approaches for feature selection we will use Information Gain for feature section process.

4) Classification

There are many data mining tools available. It can perform classification, data Pre- processing, clustering, regression, visualization and association rules.

DATASET

A data set (also known as a dataset) is a collection of information. A data set is typically defined as the contents of a single database table or statistical data matrix, where each column of the table represents a specific variable and each row represents a specific member of the data set in question. For each member of the data set, the data set lists values for each of the variables, such as an object's height and weight. Every value is referred to as a datum. The data set may include data for one or more members, with the number of rows corresponding to the number of members. The word data set

can also be used more loosely to refer to the information contained in a series of closely linked tables that correlate to a specific experiment or occurrence. Data corpus and data stock are less commonly used terms for this type of data source.

Data sets acquired by space agencies undertaking experiments using instruments aboard space probes are one example of this type. Big data refers to data collections that are so vast that typical data processing software can't handle them. The data set is the unit of measurement in the open data discipline for the information disclosed in a public open data repository. More than half a million data pieces are gathered via the European Open Data platform. Other definitions have been proposed in this sector, but there is presently no official one.

Other difficulties (real-time data sources, non-relational data sets, etc.) make reaching a consensus much more challenging. The structure and attributes of a data set are defined by a number of factors. These include the number and types of attributes or variables, as well as numerous statistical measures such as standard deviation and kurtosis that can be applied to them. Values can be numerical data (i.e., not consisting of numerical values), such as a person's height in centimetres, or nominal data (i.e., not consisting of numerical values), such as a person's ethnicity. Values might be of any of the types defined as a level of measurement in general.

The values for each variable are usually of the same kind. There may, however, be some missing values that must be communicated in some way. Data sets in statistics are frequently derived from actual observations gathered by sampling a statistical population, with each row representing observations on one constituent of that population. Algorithms can also produce data sets for the purpose of testing particular types of software. Some current statistical analysis software, such as SPSS, still displays data in a data set format. Imputation can be used to complete a data set if data is missing or suspect.

URLs	Path	Domain	Sub domain	Rank	Index
https://en.wikipedia.org/wiki/Achievements_of_Sachin_Tendulkar	1	1	1	5	1
http://blogspot.in	-1	1	1	-1	1
https://twitter.com/sachin_rt?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor	1	1	1	11	1
http://cnn.com	-1	1	1	104	1
http://alibaba.com	-1	1	1	168	1
http://odnoklassniki.ru	-1	1	1	184204	1
http://conduit.com	-1	1	1	8253	1
http://thepiratebay.sx	-1	1	1	1	1

Table 2 Data set

5. RESULTS

We conduct the simulation using the above datasets and settings. The convergence, accuracy of phishing identification, and response time are recorded as outputs of our simulation. We compare the performance of Fi-NFN to the current bench- mark approaches, such as Fuzzy [4], Neural network [1], Google PageRank [4], eMCAC [2] and FACA [1]. First, we provide a brief outline of those methods that we compare Fi-NFN with:

- Fuzzy: We compare Fi-NFN with an online algorithm that classifies URLs using the fuzzy approach. Essen- tially, Fuzzy is built by a rule set based on the URL characteristics.
- Neural network: Neural network is an approach proposed by [3] using the neural network model to identify phishing URLs. We implement a three-layer model and use our dataset to train this neural network.
- Google: This is the popular tool of Google [2][1], which can be easily installed on web browsers. Google Toolbar can detect phishing terms based on input keywords. In this work, to evaluate the performance of our approach, we develop an application that calls the Google API to detect phishing URLs at fog nodes instead of installing them on user devices.
- eMCAC [2] and FACA [1]: These are new approaches based on the rule set method to detect phishing URLs. Similar to the Google API, we implement eMCAC and FACA on fog nodes for detection.

6. CONCLUSION

In this Paper, we consider the security issues regarding the fog network to enhance network safety. In particular, we study the phishing website problem and propose identification architecture on the fog network. Based on the advantages of the fog architecture and the neuro-fuzzy approach, we propose a phishing identification model, called Fi-NFN, to protect local devices easily and quickly. Without consuming many resources from local devices, our Fi-NFN model not only transparently protects users in real time, but also improves the quality of services at the edge of the network. Without using an inefficient blacklist method, we design a five-layer neuro-fuzzy network with six heuristic input values (Primary- Domain, Sub Domain, PathDomain, PageRank, Google Index and Alexareputation). Our simulation results indicate that the efficiency of phishing identification after training with the training dataset by improving the average accuracy to 98.36% and reducing the missed detection and false alarm rates to 0.9% and 0.74 %, respectively. Simulation results show that our method is more efficient, stable and accurate. Especially, various testing results indicate that our model in a fog computing environment is not only possible, but also can be applied practically.

REFERENCES

[1] L. Wenyin, G. Huang, L. Xiaoyue, X. Deng, and Z. Min, "Phishing web page detection," in Document Analysis and Recognition, 2005. Proceedings.Eighth International Conference on. IEEE, pp. 560–564.

[2] P. Stavroulakis and M. Stamp, Handbook of Information and Communication Security, 1st ed. Springer Publishing Company, Incorporated, 2010.

[3] http://www.antiphishing.org.

[4] https://securityintelligence.com/

[5] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in Proceedings of the 16th international conference on World Wide Web. ACM, 2007, pp. 639–648.

Web References:

1. www.w3schools.com
2. www.stackoverflow.com
3. www.wikipedia.com
4. www.youtube.com