Vol. 4, No. 12 December 2013 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

An Associative Classification Data Mining Approach for Detecting Phishing Websites

[1]Suzan Wedyan, [2] Fadi Wedyan
[1]Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan
[2]Department of Software Engineering, Hashemite University, Zarka, Jordan

[1]susanwedyan@gmail.com, [2]fadi.wedyan@hu.edu.jo

ABSTRACT

Phishing websites are fake websites that are created by dishonest people to mimic webpages of real websites. Victims of phishing attacks may expose their financial sensitive information to the attacker whom might use this information for financial and criminal activities. Various approaches have been proposed to detect phishing websites, among which, approaches that utilize data mining techniques had shown to be more effective. The main goal of data mining is to analyze a large set of data to identify unsuspected relation and extract understandable useful patterns. Associative Classification (AC) is a promising data mining approach that integrates association rule and classification to build classification models (classifiers). This paper, proposes a new AC algorithm called Phishing Associative Classification (PAC), for detecting phishing websites. PAC employed a novel methodology in construction the classifier which results in generating moderate size classifiers. The algorithm improved the effectiveness and efficiency of a known algorithm called MCAR, by introducing a new prediction procedure and adopting a different rule pruning procedure. The conducted experiments compared PAC with 4 well-known data mining algorithms, these are: covering algorithm (Prism), decision tree (C4.5), associative Classification (CBA) and MCAR. Experiments are performed on a dataset that consists of 1010 website. Each Website is represented using 17 features categorized into 4 sets. The features are extracted from the website contents and URL. The results on each features set show that PAC is either equivalent or more effective than the compared algorithms. When all features are considered, PAC outperformed the compared algorithms and correctly identified 99.31% of the tested websites. Furthermore, PAC produced less number of rules than MCAR, and therefore, is more efficient.

Keywords: *Associative classification, Data Mining, Phishing Websites, Machine Learning*

1. INTRODUCTION

Financial and governmental institutes offer a variety of financial services to their clients. Online banking and online shopping become popular since the late 80's. Nowadays, almost all banks around the globe offer many online services to their clients while online shopping became a major sector of the world economy. The American Census Bureau of the Department of Commerce estimates the U.S retail e-commerce sales for the second quarter of 2013 to be about 64.8$ billion. This number is expected to increase more in the future with more retailers offering more online services [1].

A main security threat to online business comes from what becomes to be known as "Phishing Attacks". In such attacks, malicious people create webpages that mimic the webpages of legitimate websites. Clients of the legitimate site mistakenly access the faked web site and expose their financial and personal information to malicious people whom might use this information to perform illegal and criminal activities. Such criminal acts causes a lot of lose for both the clients and the legitimate companies. Moreover, phishing attacks, if continues to succeed, threatens the whole online shopping industry as a secure sector of financial activities. Several approaches have been proposed to detect phishing websites, some of which are adapted by the industry. These approaches are mainly based on keeping a list of URLs called a blacklist, such as Google Safe Browsing [2], Microsoft IE9 anti-phishing protection [3], and SiteAdvisor [4]. A blacklist is a list of URL's thought to be malicious. When a user visits a website, the browser refers to the blacklist to examine if the currently visited URL is present within the blacklist. In this case, the website is considered as malicious and the browser warns the user. The blacklist can be stored either locally (on the user's machine), or on a server that is queried by the browser for every requested URL. Blacklist approaches suffer from three main problems. First, the amount of phishing URLs available within the list. The size of the list grows up fast which increases the time required to access the list. Second, the false positive rate, which is classifying legitimate website wrongly as phishing. This has a negative influence on the user since for each false positive the user loses trust on the blacklist, and will later ignore a real warning. The third and most significant problem is timing. The effectiveness of the blacklist depends on having an up to date list. However, since most of phishing websites have a short life, if the process of updating the blacklist is slow, then there is a chance that phishing attacks can occur before being detected by the blacklist.

Another trend of approaches for detecting phishing websites relies on using a machine learning or data mining algorithm that recognize the phishing website based on a set of characteristics or features that are extracted from the website. The features are recognized by experts to be distinguishing characteristics of a phishing website (e.g., long uniform resource locater (URL), age of domain). According to these approaches, phishing is a pattern recognition problem that can be solved by chosen the "right" set of features and a "suitable" pattern discovery or recognition algorithm.

One of the recent data mining techniques is associative classification (AC) which integrates two known data mining tasks, association rule mining and classification. The classification step is added in order to use the produced classifier model for the purpose of prediction. The two data mining tasks are analogues, with the exception that classification aims to forecast the class label, while association rule describes correlations among items in a transactional dataset, Several studies (e.g., [5,6,7]) provided evidences that AC usually extracts better classifiers with reference to classification accuracy than other traditional classification approaches, such as decision trees [8], and rule induction [9].

The phishing websites problem can be viewed as a binary classification task where the output has two values phishing or legitimate. Using a classification algorithm requires building the classifier using a dataset (i.e., training) that contains a set of known websites with a target class (legitimate and phishing). A website is represented by a set of distinguishing features. Once a classifier is built, it can be used to classify websites in real time.

In this paper, an associative classification algorithm for detecting phishing websites is proposed. The proposed algorithm, called Phishing Associative Classification (PAC), is an enhanced version of an AC algorithm called Multi-class Classification based on Association Rule (MCAR) proposed by Thabtah et al. [10]. MCAR is an effective AC algorithm that has an efficient learning technique and builds an effective classifier. PAC enhances MCAR by using a novel procedure for building the classifier that cuts down unnecessary rules after finding the complete set of rules. Therefore, it's improving the efficiency of the algorithm. Efficiency, in terms of the response time, is vital for anti-phishing technique since the detection of phishing websites is a real time operation that is performed frequently (whenever a user accesses a website). PAC improves the effectiveness of MCAR by using: (1) a pruning technique that cut down unnecessary rules from the complete set of rules, and (2) introducing a new prediction procedure that uses full and partial matching instead of partial matching (as in MCAR).

In order to evaluate the effectiveness and efficiency of the proposed algorithm, an experimental study is performed. A dataset that consists of 1010 website is collected. The dataset contains 562 phishing website and 448 legitimate website. Each website is represented by 17 features categorized into 4 sets. These features were proposed by Mohammad et al. [11].The features are extracted from the website URL and contents and represent distinguishable features of phishing web-sites. PAC is compared with 4 well-known data mining algorithms. These are: C4.5 [8], Prism [12], MCAR [10], and CBA [13]. PAC has implemented using Java programming language. For the rest of the algorithm, the author of this paper used the implementation provided by WEKA [14]. Ten-fold cross-validation was used to compute the effectiveness of each algorithm.

The results on each features set show that PAC was either equivalent or more effective than other algorithms. When all features are considered, PAC outperformed the compared algorithms and correctly identified 99.31% of the tested websites. Furthermore, PAC produced less number of rules than MCAR, and therefore, is more efficient.

The rest of this paper is organized as follows. Section 2 surveys the related work proposed for detecting phishing websites. Research questions in section 3. Main concepts of associative classification are given in section 4. Details of the proposed solution are introduced in Section 5. Experimental results are discussed in Section 6. Finally, conclusion and directions for future work are outlined in Section 7.

2. RELATED WORK

In this section, the current anti-phishing approaches are surveyed, and classified into two groups. These are: Blacklist/whitelist approaches and pattern recognition approaches.

2.1 Blacklist/White list Approaches

Ludl et al. [15] measured the effectiveness of two popular blacklist based approaches. These are: the blacklist preserved by Google and used by Firefox, and the blacklist preserved by Microsoft and used by Internet Explorer. Their results show that Google was able to label 90% correctly, but Microsoft labels only 67%.

Sharif et al. [16] proposed a phishing blacklist approach that avoids the problem of keeping the blacklist up to date. Their proposed approach can be installed on the mail server to identify the set of URLs in an email, and the attacked company name. The authors have conducted an experiment to contrast the URLs collected from the email with that of the actual company obtained from Google search engine. The results show that their approach can score about 100% accuracy in detection phishing URLs with 9% of false positive. Though, the authors did not show how they can get the logo of the companies worldwide or how image comparison was performed. Furthermore, they did not show how to deal with URL address that is hidden by a proxy which limits the practicality of their study.

Sheng et al. [17] revealed that blacklists are updated at various speeds. They estimated that 47% - 83% of phishing URLs are added to blacklists 12 hours after they lunched. Moreover, the authors found that zero hours protection delivered by major blacklist-based toolbars claims a true positive between 15% and 40%. So it is necessary for a decent blacklist to be updated instantly.

The opposite term to blacklist is whitelist, which is a set of trusted websites, while all other websites are considered bad or untrusted. Chen and Guo [18] proposed

an anti-phishing approach called Automated-Individual-Whitelist (AIWL), based on an individual user's whitelist of known trusted sites. AIWL trace every login attempts performed by the users individually using a Naive Bayesian classifier. In case a repeated successful login for a specific website achieved, AIWL prompts the user to add the website to the whitelist. Users are warned once they submit their credentials to a website that does not exist in the whitelist. This technique assumes that users solely repeatedly submit credentials to legitimate sites, however all other sites are considered malicious.

Afroz and Greenstadt [19] proposed an approach called PhishZoo that uses whitelist and blacklist, PhishZoo builds profiles of trusted websites based on fuzzy hashing technique. A profile of a site is a combination of different metrics that uniquely identifies that site. This approach combines the ability of whitelisting approaches to detect new or targeted phishing attacks with the ability of blacklisting and heuristic approach to warn users. The authors believe that phishing detection should be from user's point of view since over 90% of users depend on websites appearance to verify its authenticity. The main hypothesis raised in this paper was, "looking-content those are: images, HTML code and scripts, can be extracted from a website automatically to build any website profile". PhishZoo evaluated using 636 phishing sites and 20 profiles of legitimate sites downloaded from phishtank [20]. The first experiment was taken to verify how many phishing sites reuse the exact or very similar html code of the real site. Only the html code of a site was considered in the profile content, the results show that 49% of phishing websites can be detected using only HTML code. However, if the logo of a site is added to the profile content then the prediction rate will increase 54%. The second experiment taken was by applying fuzzy hashing technique to separate content elements, i.e. images, html codes, and scripts. Matching threshold play an effective role in detecting a phishing website. When the threshold was set to 0.2, the prediction accuracy was 82%, but if the threshold was set 0.3, PhishZoo gives an accuracy of 67%. The main drawback of this approach was in step 6 mentioned above since PhishZoo claims that most phishing websites are simply copies of real sites. But if the loaded site which could be a phishing website does not look like the real website it is imitating and that could occur just by changing the size or the position of the site logo, then PhishZoo will ask the user to judge on the legitimacy of the loaded website. The user will also be asked to build a new profile for that website.

2.2 Pattern Recognition Approaches

Abu-Nimeh et al. [21] presented a study that compares the effectiveness of six machine learning approaches in detecting phishing emails. These are: Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet). The authors collected a data set that consists of 1171 raw phishing emails and 1718 legitimate emails. Each email is represented by 43 features while effectiveness is measured by a weight error term which gives a higher weight to false negatives than false positives. This approach of measuring effectiveness is widely used for spam filter because the effect of considering a legitimate email as spam is worse than letting a spam email pass to the client mailbox. The authors applied a weighted error measure that considers a false positive 9 times more costly than a false negative. The results show that LR has the lowest error rate of 3.82% while RF has the highest of 5.78%.

Garera et al. [22] proposed an approach for detecting phishing URLs using logistic regression. A website is represented using 18 manually selected features. Their approach achieved a classification accuracy of 97.3% over a set of 2,500 URLs collected by Googles blacklist of URLs for Firefox.

Fette et al. [23] proposed an approach that uses statistical methods in machine learning to classify phishing emails. Their classifiers examine 10 features that describe the URL itself and the contents of the email (e.g., the number of URLs, number of domains, and number of dots in a URL). The authors used random forest as a classifier. Random forests create a number of decision trees and each decision tree is made by randomly choosing an attribute to split on at each level, and then pruning the tree. Their approach was evaluated using a data set that consists of 860 phishing emails taken from Spam Assassin and phishing corpus repositories and 6950 legitimate emails taken from phishing corpus. The classifier correctly identified 96% of the phishing emails with less than 0.1% false positives.

Ma et al. [24] used statistical methods for classifying site reputation based on the relationship between URLs and the lexical and host-based features that characterize them. The authors used three classifiers: Nave Bayes, Support Vector Machine (SVM), and logistic regression. Their results show that the classifiers obtained 95% accuracy (where LR has the highest) while maintaining a very low false positive rate.

Dunlop et al. [25] proposed an approach called GoldPhish that can detect zero-day phishing attacks (i.e., new phishing attacks). Phishing websites usually last for few days and sometimes for only few hours, therefore the authors proposed GoldPhish which operates in three steps, in the first step it captures an image of the current website, and then in the second step, the captured image is converted into text which is given as input to the third step into which Google's search engine is utilized to retrieve the search result URL's. GoldPhish only uses the first four results to decide whether a site is legitimate. This is because legitimate websites will generally come up in the first results because of the page rank technique used by Google.

Liu et al. [26] proposed an approach that identifies clusters of web-pages that are associated to the suspicious webpage. The features used to identify associated web-pages include link relationship, ranking relationship, text similarity, and webpage layout similarity relationship. A DBSCAN clustering method is employed to find if there is a cluster around the suspicious webpage. If such cluster exists, the webpage is considered as a phishing webpage and then find its phishing target from this cluster. Liu et al. results show that the approach successfully identified 91.44% of their phishing targets and a false positive rate of 3.40% was recorded.

Aburrous et al. [27] applied association rule and classification data mining algorithms for predicting phishing sites. The authors used 20 features which they classified into 6 groups. These are (1) URL and Domain identity, (2) security and encryption, (3) page style and contents, (4) Web address bar, (5) social human factors, and (6) Source code and Java Script. And used three fuzzy set values to describe each site (Genuine, Doubtful and Legitimate), the output target attribute has the following a set of possible values (Very Legitimate, Legitimate, Suspicious, Phishy, Very Phishy). They conducted an experiment using the following classification data mining techniques: JRip, PART, PRISM and C4.5, and associative classification (CBA, MCAR). They used a data set that consists of 412 phishing e-banking websites, 288 suspicious and 306 of real e-banking web-sites. The result shows that associative classifiers are more accurate than traditional classification algorithm; MCAR outperform all other traditional classification in term of accuracy and speed and it generates 22 classification rules.

Basnet et al. [28] proposed a rule-based approach for detecting phishing websites. Their approach was evaluated using 16,797 phishing websites from PhishTank [20] repository and 24,086 legitimate websites taken from Yahoo directory [29], and DMOZ [30]. The rules were then used as features in Decision Tree and Logistic Regression learning algorithms and their performance results were compared. C4.5 and LR gave competitive accuracy of 99% and FPR of 0.5% and FNR of 2.5%. Their performance slightly degraded, however, when tested with new data sets against models trained with old data set.

3. RESEARCH QUESTIONS

This article aims to answer the following research questions:

a. How good is the proposed PAC algorithm in predicting phishing websites? Phishing website is a classification problem that requires the analysis of large amount of data. AC algorithms have been used to solve similar problems. Therefore, PAC can be a good candidate to solve the problem.
b. How good is the PAC algorithm compared with other data mining algorithms in predicting phishing websites? The PAC algorithm is compared with 4 well-known data mining algorithms. These are: C4.5 [8], Prism [12], MCAR [10], and CBA [13]

4. ASSOCIATIVE CLASSIFICATION

In general, an AC algorithm works in three steps. Step one, discovering and generating the rules. Step two, building the classifier, and prediction in step three. In the rule discovery step, the frequent rule items are discovered and the complete sets of rules are generated as “Class Association Rule” (CARs). After that, the rules are ranked according to certain threshold parameters such as confidence and support values. In step 2, the complete set of CARs are filtered and pruned to remove duplicated and useless rules, since the number of rules generated run into several thousands, and furthermore many of them are both redundant and not discriminative among the classes during building the classifier. The remaining rules are selected to represents the final classifier. Finally, in prediction, the classifier derived gets tested on new independent data set to measure its effectiveness in forecasting the class of unseen test cases. The prediction accuracy is the percentage of correctly classified test cases in the test data set.

AC algorithms depend on two important thresholds: minimum support and minimum confidence. Minimum support (*MinSupp*) represents the frequency of the attribute value and its related class (attributes, class) in the training data set. Minimum confidence (*MinConf*) represents the frequency of the attribute value and its related class in the training data set (< attributes, values >, class) from the frequency of that attributes value in that training data.

In AC mining MCAR [10], the authors suggested using vertical layout of the association rules discovery algorithm proposed by Zaki et al. [31] in order to reduce the computational time needed to produce the rules. MCAR algorithm modified the Tid-list intersection learning approach used in association rule to find the set rules CARs from the training data set, which only stores the differences in the transactions identifiers (TIDs) of a candidate rulitems from its generating frequent rulitems. The rule ranking procedure of MCAR is based on rule confidence, support, the number of attributes in the rule antecedent, and the class distribution in the training data set as a tie breaking condition. In particular, if two rules have identical confidence, support, and antecedent length, MCAR favors the rule which is associated with the class that has larger frequency in the training data set. Once the complete sets of rules are found and ranked, subset of highly effective rules is chosen to represent the classifier.

There are different pruning methods use in AC to build the classifier, for instance, MCAR uses Database coverage which considers a rule as significant if its body fully matches the training case and the rule has a common class with the training case class. If so, the rule gets inserted into the classifier. In cases the rule body does not

match any training case then the rule is discarded. The last step in the life cycle of MCAR is to allocate the appropriate class to test cases, which is class prediction. MCAR iterates over the rules in the classifier and assigns the class associated with the highest sorted. If there are no rules match the test case body, MCAR takes a default class and assigns it to the test case.

5. THE PROPOSED APPROACH

The proposed algorithm (PAC) targets to solve the phishing websites problem and also adds the following enhancements over the MCAR algorithm.

a. Ranking based on minority class distribution among rules when two or more rules having the same confidence, support and rule length during the process of rule ranking.
b. A novel pruning procedure that cuts down unnecessary rules after finding the complete set of rules during the classifier building.
c. A new class assignment method that considers full and partial match procedure to give the test data the right class during prediction.

PAC deals with the above mentioned enhancements. It also takes advantage of the vertical learning which iterates over the data set only once, solving an important problem in data mining. Normally, AC algorithms like CBA [13] and LC-AC [7] the rule generation phase is performed in an iterative manner so the joining of frequent itemsets of size 'k' to generate candidate itemsets of size 'k+1', is carried out in which each k -itemsets found necessitate a full scan over the data set. The above process of merging is computationally expensive and requires substantial CPU time. At the same time it produces a large number of candidate itemsets in each iteration due to unnecessary itemsets joining. The proposed data representation method of PAC used the MCAR approach which enhances the process of discovering frequent features values (items) of these algorithms by using vertical format and Tids-list intersections, which enables the rules discovery without repetitive scans that necessitate high demand on resources including training time during rule discovery step in items support calculation. One main focus of this paper is to minimize the cost associated with the time during frequent ruleitemsets generation process of these algorithms.

5.1 Preprocessing

PAC uses attribute that take discrete values, since some of the features take continuous values (e.g., age of the domain) then discretization of these features is necessary to be able to use them with PAC. For categorical attribute, all the possible values are mapped to a set of integers. The process of discretization starts by selecting a continuous attribute from the training data and sort it in ascending order with the class values of each instance [32]. A break point is placed where the class value changes. Information gain measure is calculated for all break points. The information gain depicts the quantity of information required to specify the class values. The break point is selected which minimizes the information gain over all other break points. The process starts again on the lower range of that attribute.

5.2 Rule Discovery

The proposed algorithm employs fast rule learning method based on vertical mining concept that utilizes simple intersection among item IDs to find the rules. PAC starts scanning the training data set to discover the frequent 1-ruleitems that hold enough support. Once all frequent 1- ruleitems are found and their occurrences in the training data (rowIDs) are determined then they stored in an array in a vertical format. Also, classes and their frequencies are stored in an array. Any ruleitems that fails to pass the *MinSupp* threshold is discarded. The proposed algorithm employs the sets of TIDs of any frequent items of size N-1 to discover the possible frequent items of size N during the rule discovery step. The result of an intersection among the TIDs of two items gives a new TIDs list, which has the locations where both items occur together in the input data. This new TIDs list can be used to calculate the support and confidence of the new item resulted from the intersection. In other words PAC finds frequent 1-items after scanning the training data set once. Then, during the discovery of frequent items of larger size it simply intersects the TIDs of the disjoint 1-items to discover the candidate 1-items of size 2 and after obtaining frequent 2-items, the candidate frequent 3-items can be derived by intersecting the TIDs of disjoint 2- items, and so on. Once the complete set of frequent items are discovered, the computation of their confidence values is a straight forward process. In fact, using the set IDs of items one can determine whether the frequent item can be converted into rules by contrasting the items's confidence value with the user minimum confidence. In a vertical mining each item in the original training data set is converted into ColumnId and RowId representation. This representation holds information about items frequencies in the training data set which later can be useful in computing the support (frequency) of an item easily starting from items of size"1".

Consider for example the following vertical data representation given in Table 1. Assume that *MinSupp* and *MinConf* are set to 20% and 35%, respectively. In the first scan, the frequent 1- ruleitems that pass the *MinSupp* threshold are discovered (e.g. <At1, a>, <At1, b>, <At2, d>, <At2, e>, <At2, f>) and the other infrequent item is discarded <At1, c>.For instance, if we take the frequent 1-ruleitem for < (At1, a) >, < (At2, e) > which their frequencies are represented by the following TIDs lists {2, 3, 4, 5, 6, 9} and {2, 4, 5} respectively. The new 2-item for < (At1, a), (At2, e) > can be determined by performing intersection by their locations in the TIDs lists. The resulting set {2, 4, and 5} denotes the TIDs where both items appear in the training dataset with class2. Now if the new potential rule < (At1, a) (At2, e), C2 > has sufficient support greater than the *MinSupp* then calculate its confidence. If the rule survives the *MinConf* parameter, it is considered as a potential rule. In the example, < (At1, a), (At2, e), C2> has a support and

confidence of 3/10, 3/3, respectively. Therefore, it is considered as a candidate rule.

Table 1: Vertical training data representation

(At1,a)	(At1,b)	(At1,c)	(At2,d)	(At2,e)	(At2,f)	(Class1,C1)	(Class2,C2)
2	1	10	6	2	1	1	2
3	7		7	4	3	6	3
4	8		9	5	8	7	4
5			10			8	5
6						9	
9						10	

5.3 Rule Ranking Procedure

AC algorithms often generate a large number of rules, which decreases the algorithm's efficiency. Keeping a smaller number of rules is vital for PAC since the algorithm is going to solve a real-time problem. Rule ranking is the first step to build a classifier in AC and it is mainly utilized to choose the most useful rules for prediction. PAC ranks rules on the basis of the following criteria: first, rule confidence, if there are two or more rules have the same confidence then rule support is considered. If they have same support then, rule antecedent length (the rule has fewer conditions in its left hand side) is considered. Finally, if rules have the same confidence, support and length, PAC chooses the rule that associate with the minority class distribution in the training data set.

PAC has enhanced the rule ranking criteria used by the MCAR algorithm to select minority rules which have less representation in the training data. This ranking method balances the generation of rules according to class label since more representative classed are already had rules generated. While the rule sorting process of MCAR considers majority class when two or more rules have similar confidence, support and length cause the generation of unbalanced classifier.

5.4 Classifier Construction

After the complete set of rules are found from the training data and got ranked, PAC algorithm evaluates the rules to come out with the most significant ones for building the final classifier for prediction, PAC iterates over all potential set of rules and marks the first one that matches the training case as a classifier rule, and the training case that covered by the rule are deleted. This process is repeated until all training cases are utilized or when there are no more rules to be evaluated, the outputs of all marked rules are used to construct the final classifier. During the evaluation steps, PAC tests the rule with the training data case based on the matching between the rule body and the training data case regardless of the class label correctness, in order to come out with the procedure that has the least negative effect on the rules.

Moreover, to reduce the over-fitting of the final classifier and decreases the size of it as well, PAC pruning method first checks if a training case has a full match with the rule. If so, the rule is given to the classifier, and all its associated covered training dataset are deleted. PAC utilized this pruning method based on its advantages on the final classifier achieved by previous research studies on AC. The pruning method of PAC is given in Figure 1.

Input: Training data set and the generated Ranked Rules
Output: Classifier

a: For each rule starting with the first ranked rule do
b: Find all applicable training data cases that match with the selected rule body and mark them
c: If the rule covers at least one training data case, insert the rule to the classifier and remove all training cases that covered by the rule.
d: If the selected rule does not cover any training case then discard it.
e: Repeat all the previous steps until the training data is not empty yet or the algorithm did not pass all rules
f: If the training data gets empty or the algorithm has passed all over generated rules, all the unmarked rules gets discarded and the marked rules gets generate as a classifier.
g: Uncovered training data cases represent a default class rule for the majority class among them (highest frequency class).

Fig 1: PAC Pruning Method

There are different pruning procedures used in AC algorithms for building the final classifier. Database coverage algorithm is the first pruning method proposed by CBA [13], where rules that cover correctly a certain number of training cases are marked as accurate rules. Several AC algorithms have successfully employed database coverage in building the classifier, such as MCAR [10] and ACCF [33].

One notable difference between PAC pruning and that of CBA or MCAR is that CBA and MCAR rule pruning methods require all items in the candidate rule and the class label be contained in the training case during evaluation in order to consider the rule as significant. In other words, if the class label in the rule does not match the class label in the training case, CBA or MCAR does not take this into consideration. The proposed algorithm considers only the full match between the candidate rule body and the training case and marks candidate rule as significant when this happens.

5.5 Prediction Procedure

Prediction is the final and most important step in classification that allocates the appropriate class to test cases. There are to main methods for class prediction in AC algorithms. In the first method, the highest ranked rule in the classifier to predict the test case class (e.g., MCAR [10]). In the second method, multiple rules are used to allocate the test case class (e.g., CMAR [34]).

PAC has introduced a new prediction procedure contains full and partial match. The new hybrid prediction approach starts with fully match to classify a test case (t) using classifier (R), PAC selects the rule in the classifier that its antecedent (body) identical matches the test case body. The fully matching works as follow:

- If the test case (t) fully matches a rule (r) in the classifier then PAC assigns the class label of the rule to test case (t).
- If the test case (t) fully matches with more than one rule in classifier then PAC assigns the first rule which have the highest confidence.

Otherwise, if there are no rules fully match the test case body, PAC considers the partial matching by selecting the rule that its body partially matches the rule's body. Each rule in the classifier that partially matches the test case is given a weight which represents the number of corresponding values (items) between the test case (t) and the rule (r) over the total number of the example's items. The weight of the rule can be computed according to equation 1:

$$weight(r,t) = \frac{m}{n} \qquad (1)$$

Where,
r: the rule
t: the test case
m: is the number of corresponding items between rule *(r)* and the test case *(t)*
n: is the total number of the example's items

The rules that their weights pass a predefined minimum weight are ranked in a descending order. Then the test case is given the class label of the rule (r) that has the highest weight, if two or more rules have the same highest weight then the class label of the rule that hold highest confidence is assigned to the test case. Finally, in cases when no rules in the classifier are applicable to the test case, the default class will be assigned to that case, which is majority class in the training dataset

Table 2: Classifier Model Example

	Right Click	Redirect Page	Pop-up Window	On Mouse Over	Class
r1	true	low	low	low	Legitimate
r2	true	low	high	high	Legitimate
r3	false	high	low	low	Phishing
r4	false	high	high	high	Phishing

Suppose we have a classifier model that consists of 4 rules as given in Table 2, and we want to predict the class labels of the following test cases below:

case1: (true, low, low, low) →??

case2: (false, high, low, high) →??

Based on the above model, case 1 fully matches the body of r1. Consequently, the class label of case 1 is Legitimate which is the class label of r1. However, case 2 does not fully match with any of rules in the model so PAC considers partial matching for this case. Case 2 corresponds with r1 and r2 into one item (low) and (high), respectively. The weight of the two of them is ¼. In r3 and r4 case 2 corresponds into three items. The weight of r4 and r3 is ¾ which is the highest weight of the other rules. The class label of r3 and r4 is Phishing, so the class label Phishing is given to case 2. Suppose the two rules r3 and r4 have different class label Phishing and Legitimate respectively, in this case PAC choses the class label of the rule which its confidence greater than the other.

We can note that the prediction procedure of PAC not only uses full match single rule prediction like CBA-based algorithm [13], or multiple rules prediction like CMAR-based algorithm [34] or partial match single rule like MCAR-based algorithm [10]. PAC takes advantage of using hybrid prediction approach that takes into account full match and partial match prediction.

6. EXPERMINTAL RESULTS

The empirical study is a vital step in order to verify the accuracy of a proposed solution. In this Section, the performed experiments are described, including the

collected dataset (Section 6.1), features assessment and selection (Section 6.2), tool implementation (Section 6.3). Finally, the results of the study are presented in (Section 6.4).

6.1 Data Collection

A set of 1010 websites, 562 phishing websites were collected from Phishtank archive [20]. PhishTank is a free community site where users can submit, verify, track and share phishing data. In addition, 448 legitimate websites were collected from yahoo directory [29] and starting point directory [35]. Both directories contain addresses of legitimate websites for different types of services.

6.2 Features

In this paper a set of features are used suggested by Mohammad et al. [11]. The set consists of 17 features and are extracted automatically using a JavaScript and PHP script. The tool also performs discretization of the selected features. The features are categorized into four sets. These are:

a. Address Bar Based Features. Features in this set are extracted from the address-bar of a website, by using a JavaScript program. The features are: (1) IP address, (2) Length of the URL, (3) whether the address contains @ symbol, (4) whether the prefix or suffix contains the dash symbol (-), (5) number of sub-domains, and (6) whether the website uses HTTPS and SSL Certificate.
b. Abnormal Based Features. PHP script is used for extracting those features. This set contains features related to the hostname in URL or revolved from the IP address of the website. This set contains four features. These are: (1) whether the resources of the websites (e.g., images, scripts) are located within their own domain, (2) the use of internal links, and (3) Server form handlers having empty string () or different domains are usually suspicious (4) and Abnormal URL.
c. HTML and JavaScript Based Features. This set contains features that are extracted from the HTML tags and JavaScripts in a website, by using JavaScript program. The set contains four features. These are: (1) Redirect Page, (2) using onMouseOver, (3) disabling Right-Click, and (4) using PopUp Window.
d. Domain Based Features. These features are extracted from WHOIS database [36] and from Alexa.com [37] by using PHP script. The set contains three features. These are: (1) age of domain, (2) DNS record, and (3) website traffic.

6.3 Tool Implementation

PAC is implemented in a tool using Java programming language. The tool has a GUI that allows the user to set the different algorithm parameters and run the experiments. The tool reads the training files given in text format where each row represents a set of features for a website. The rules generated by the PAC algorithm can be saved to a file and also displayed in the GUI. The tool can perform cross validation for assessing the results.

6.4 Results

The proposed algorithm is tested using the collected data set. Ten-fold cross-validation is used to compute the accuracy of the algorithms. Four popular classification algorithms have been compared with the PAC algorithm in terms of classification accuracy, each of the four algorithms utilizes different methodology in producing knowledge. These algorithms are:

a. Decision tree C4.5 [8], which is a popular decision tree algorithm that utilizes a divide and-conquer methodology for extracting knowledge. This algorithm starts by choosing the best attribute as a root node, where each branch of the root corresponds to one of its possible value.
b. PRISM [12], which is a common covering algorithm, that utilizes a recursive greedy approach based on the distribution of class labels in the training data. Prism normally generates perfect rules (those with 0% error rate) and measures the accuracy of its rules using the accuracy formula: (P/T).Where P represents the number of positive examples and T represents the number of negative examples covered by a rule.
c. Classification Based on Association algorithm (CBA) [13]. CBA is an associative classification algorithm that utilizes the Apriori algorithm [38]. The Apriori algorithm discovers the frequent items through multi scans over the training data set.
d. Multi-class Classification based on Association rule (MCAR) [10]. MCAR is an Associative Classification algorithm that utilizes Tid-list intersection.

The experiments of C4.5 and PRISM were conducted using the Weka software system [14]. WEKA stands for Waikato Environment for Knowledge Analysis. WEKA is an open Java source code for the machine teaching community that includes implementations of different methods for several different data mining tasks such as classification, clustering, association rule and regression. CBA experiments were conducted using a VC++ implementation version provided by Liu et al. [13]. The *MinSupp* is set at 5% and *MinConf* to 35%, as in [10] for both PAC and CBA

The performance of the algorithms is measured using the classification accuracy metric. Accuracy is computed by the percentage of correctly classified websites in the test data set. In order to minimize the effect of the different attributes on the results, accuracy of each of the algorithms is measured using each features category as well as using all of the features.

Figure 2 shows the classification accuracy for each algorithm using Abnormal based dataset. The results show that the PAC algorithm scored higher classification accuracy than C4.5, MCAR, CBA, and Prism. The results also suggest that the Abnormal based features are strong since all the algorithms (except Prism) scored classification accuracy over 90%.

Figure 3 shows the classification accuracy for each algorithm using Address Bar based dataset. The results show that the PAC and C4.5 algorithm scored equivalent classification accuracy, then MCAR, CBA, and Prism. The results also suggest that the Address Bar based features are strong since all the algorithms (except Prism) scored classification accuracy over 90%.

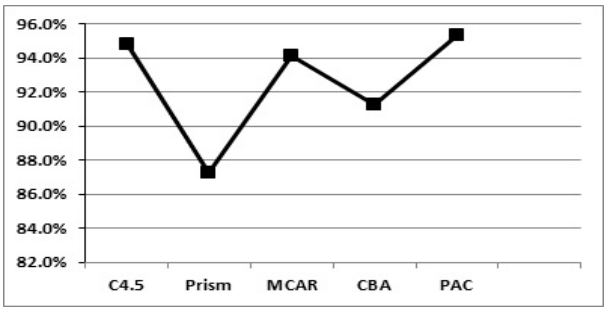

Fig 2: Classification accuracy using Abnormal Based Features

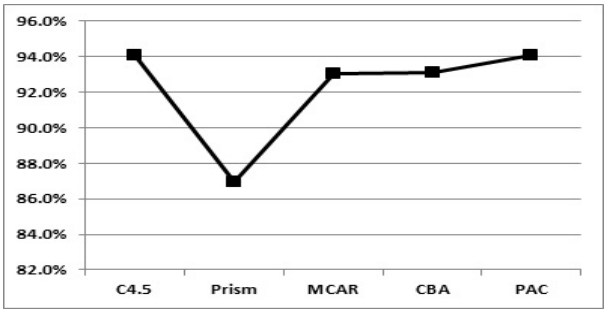

Fig 3: Classification accuracy using Address Bar Based Features

Figure 4 shows the classification accuracy for each algorithm using Domain based features. The results show that the PAC algorithm scored slightly higher classification accuracy than C4.5, MCAR, CBA, and Prism. Domain based dataset, when used alone, are not strong in recognizing phishing websites. None of the algorithms reached 90% classification accuracy.

Figure 5 shows the classification accuracy for each algorithm using HTML and JavaScript based dataset. The results show that the PAC and MCAR algorithm scored equivalent classification accuracy. The accuracy of the two algorithms was higher than C4.5, CBA, and Prism. Among all features sets, HTML and JavaScript based features, when used alone, are the strongest where all algorithms, except Prism, scored classification accuracy over 96%.

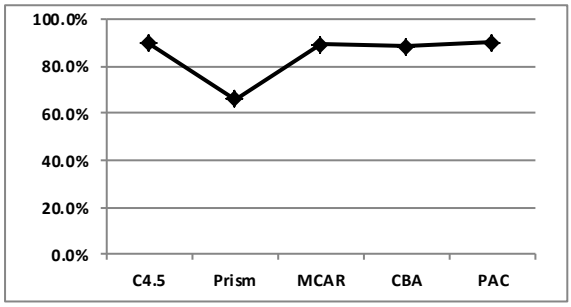

Fig 4: Classification accuracy using Domain Based Features

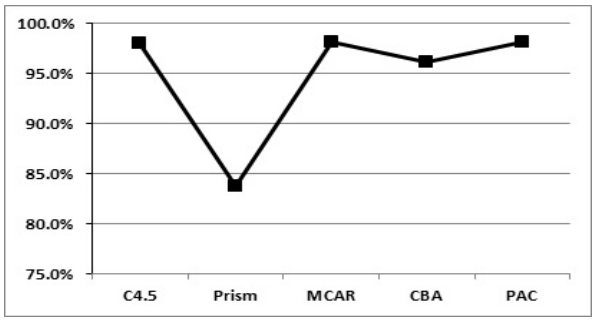

Fig 5: Classification accuracy using HTML and JavaScript Based Features

Figure 6 shows the classification accuracy for each algorithm using all features. The results show that PAC outperforms all other algorithms with a classification accuracy of 99.31, Note also that all algorithms scored classification accuracy over 98%. The Prism algorithm, which scored low accuracy in each set of features, was able to score an accuracy of 99.11% by gathering all datasets (higher than CBA and equivalent to C4.5). This indicates that Prism was able to find the rules that distinguish phishing websites from legitimate, when given enough data. Further, when all features are used, it's noticed that all of the algorithms generated rules in which the Domain based features, especially the age of domain and web traffic features, were strong in detect phishing website. While as our previous results show, these features are weak when used alone.

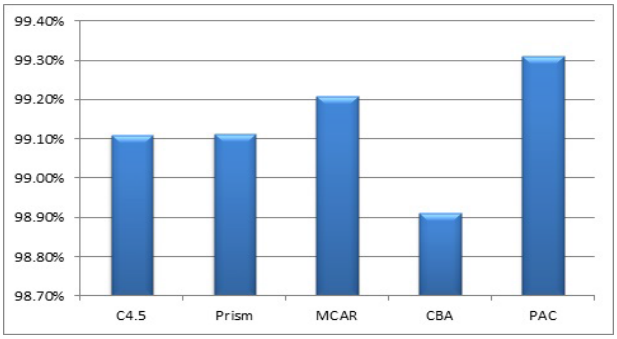

Fig 6: Classification accuracy of C4.5, Prism, MCAR , CBA and PAC using all features

In order to measure the efficiency of the proposed algorithm, the number of rules produced by PAC is compared with the number of rules produced by the MCAR, the original algorithm by using each set of features and all features. The same values of 5% and 35% are used for both algorithms for *MinSupp*, *MinConf*, respectively.

The results for comparing the number of rules are shown in Figure 7. As shown in the figure, PAC has produced less number of rules for Abnormal based features, Address bar based features, Domain based features, and all features. Only in the HTML and JavaScript based features, both algorithms produced the same rules (and have the same effectiveness).

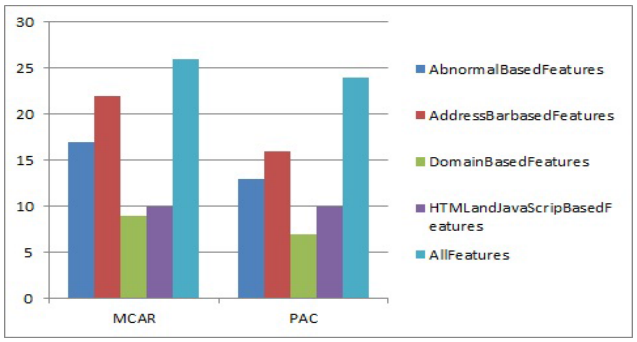

Fig 7: Number of Rules produced by MCAR and PAC algorithms

In this paper, we proposed a new associative classification algorithm called PAC "Phishing Association Classification". PAC employed a new prediction procedure that improves the prediction rate of the resulting classifiers by using this hybrid approach that considered full and partial matching in class assignment. Moreover PAC adopted a pruning procedure in constructing the classifier which results in generating moderate size classifiers. The algorithm improves the effectiveness of and the efficiency of the MCAR algorithm.

The algorithm was applied to solve the phishing website problem. The conducted experiments compared PAC with 4 well-known data mining algorithms. Experiments were performed using 4 sets of features, in which PAC has outperformed in Abnormal based features and Domain based features, PAC and MCAR scored the same classification accuracy in the HTML and JavaScript features. In Address bar based features PAC and C4.5 scored the same classification accuracy, while using all features, all the algorithms scored high classification accuracy, but PAC scored the highest.

The results show the importance of using suitable features and also the effect of combing the features on the classification algorithm. The Prism scored a poor classification accuracy using each feature set. However, when the features are combined, Prism was able to find the relation between these features that distinguish phishing websites from the legitimate ones. While PAC was able to find suitable rules using all each features set, the algorithm took advantage of using 17 features and found a compact set of rules that are able to classify the websites with high accuracy.

7. CONCLUSIONS AND FUTURE WORK

The proposed work can be extended in many directions. These include:

a. Applying the algorithm on other classification problems. The phishing websites is a binary classification problem. The algorithm can be evaluated with other problems with similar characteristics (e.g., Spam emails).
b. Investigating different types of features. PAC scored high classification accuracy with the set of features proposed by Mohammad et al. [11]. However, since phishers tend to change their techniques rapidly, it's possible that these features lose their value over time. Therefore, the algorithm can be evaluated with other features and measure the impact on its accuracy.
c. Investigating the scalability of the algorithm. Scalability measures the ability of a solution to deal with large scale problems, without losing its accuracy. This is an important attribute for any deployable solution.
d. A large empirical study with different data sets can be performed to confirm the obtained results. Data sets can be obtained from other corpus.

REFERENCES

[1] U.S Census Bureau, "quarterly Retail E-Ccommerce Sales 2nd Quarter 2013," http://www.census.gov/retail/mrts/www/data/pdf/ec current. pdf, January 2013.

[2] Google Safe Browsing, https://developers.google.com/safe-browsing/, January 2013.

[3] Microsoft phishing filter, http://www.microsoft.com/, January 2013

[4] McAfee Site Advisor, http://www.siteadvisor.com/, January 2013

[5] E. Baralis, S. Chiusano, and P. Garza, "A lazy approach to associative classification," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 2, pp. 156–171, 2008.

[6] Q. Niu, S.-X. Xia, and L. Zhang, "Association Classification Based on Compactness of Rules," in WKDD 2009. Second International Workshop on Knowledge Discovery and Data Mining, 2009, pp. 245–247

[7] F. Thabtah, Q. Mahmood, L. McCluskey, and H. Abdel-Jaber, "A New Classification Based on

Association Algorithm," Journal of Information & Knowledge Management, vol. 9, no. 01, pp. 55–64, 2010

[8] J. R. Quinlan, C4. 5: programs for machine learning. Morgan kaufmann, 1993, vol. 1.

[9] D. Jensen and P. R. Cohen, "Multiple comparisons in induction algorithms," Machine Learning, vol. 38, no. 3, pp. 309–338, 2000.

[10] F. Thabtah, P. Cowling, and Y. Peng, "MCAR: Multi-Class Classification Based on Association Rule," in The 3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005, pp. 1–21.

[11] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in IEEE International Conference for Internet Technology and Secured Transactions, 2012, pp. 492–497.

[12] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," International Journal of Man-Machine Studies, vol. 27, no. 4, pp. 349–370, 1987.

[13] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'08), 1998, pp. 80–86.

[14] WEKA, Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/, December 2012.

[15] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in Detection of Intrusions and Malware, and Vulnerability Assessment, 2007, pp. 20–39.

[16] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," in Computer Systems and Applications, 2008.Conference on. IEEE, 2008, pp. 840–843.

[17] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in Sixth Conference on Email and Anti-Spam (CEAS), 2009.

[18] J. Chen and C. Guo, "Online detection and prevention of phishing attacks," in ChinaCom'06. IEEE First International Conference on Communications and Networking in China, 2006, pp. 1–7.

[19] S. Afroz and R. Greenstadt, "Phishzoo, "Detecting phishing websites by looking at them," in Fifth IEEE International Conference on Semantic Computing (ICSC), 2011, pp. 368–375.

[20] PhishTank, http://www.phishtank.com/, October 2012.

[21] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, 2007, pp. 60–69.

[22] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in Proceedings of the 2007 ACM workshop on Recurring malcode, 2007, pp. 1–8.

[23] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in Proceedings of the 16th international conference on World Wide Web, 2007, pp. 649–656.

[24] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: an application of large-scale online learning," in Proceedings of the 26th ACM Annual International Conference on Machine Learning, 2009, pp. 681–688.

[25] M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: Using Images for Content-Based Phishing Analysis," in IEEE Fifth International Conference on Internet Monitoring and Protection (ICIMP), 2010, pp. 123– 128.

[26] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," in IEEE 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 4153–4156.

[27] M. Aburrous, M. Hossain, K. Dahal, and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies," in ITNG: Seventh International Conference on Information Technology: New Generations, 2010, pp. 176–181.

[28] R. B. Basnet, A. H. Sung, and Q. Liu, "Rule-based phishing attack detection," in International Conference on Security and Management (SAM), 2011, pp. 373–?383.

[29] Yahoo Directory, http://dir.yahoo.com/. November 2012.

[30] DMOZ Open Directory Project, http://www.dmoz.org. November 2012.

[31] M. J. Zaki and K. Gouda, "Fast vertical mining using diffsets," in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003, pp. 326–335.

[32] U. Fayyad and K. Irani, "Multi-interval discretization of continuous valued attributes for classification learning," in Proceedings of the 13th International Joint Conference on Artificial Intelligence,

[33] X. Li, D. Qin, and C. Yu, "ACCF: Associative classification based on closed frequent itemsets," in FSKD'08. IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 2, 2008, pp. 380–384.

[34] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in ICDM 2001: Proceedings IEEE International Conference on Data Mining, 2001, pp. 369–376.

[35] Starting Point Directory, http://www.stpt.com/directory/. November 2012.

[36] WhoIS, http://www.who.is.com /. November 2012

[37] Alexa the Web Information Company, http://www. alexa.com/. November 2012.

[38] R. Agrawal and R. Srikant, "Fast algorithms for mining association rule" in Proceedings of the 20th International Conference on Very Large Data Bases Santiago, Chile, 1994, pp. 487-499.