Web usage Anomaly Detection through User Profiling with Naïve Bayes Classifier

Gregg Victor Gabison[1], Bobby D. Gerardo[2,] Elmer Maravillas[3]

[1]University of San Jose – Recoletos
gregg@usjr.edu.ph
[2]West Visayas State University
bgerardo@wvsu.edu.ph
[3]Cebu Institute of Technology University
elmer.maravillas@gmail.com

Abstract. The objective of this paper is to develop a web application that will detect possible anomaly in the web usage through user profiling with the use of the naïve bayes classifier. In the preprocessing activity, the two repositories namely: web activity logs and user log-ins are integrated as one data source set which will serve as the corpus of this application. For the training process, we construct and analyze the visited sites in terms of n-grams, resulting to an idiom agnostic and trigram form. Using the Naïve Bayes Classifier, we adopt the Multinomial model where it captures the frequency of words, not just their presence as compared in the Bernoulli model. Comparing the generated user profile/ class, a likelihood score is computed determining its similarity vis-à-vis the new user web activity. In validating the generated likelihood scores, a confusion result matrix is presented showing the accuracy of the generated scores.

Keywords: Web Data mining, data mining, web log analysis, pattern detection, Naïve Bayes, machine learning, profiling, classification, anomaly detection

1 INTRODUCTION

Today, in any organization, one of the fundamental service extended to its employees and stakeholders, is the provision of internet access. However such service has to be managed, monitored and secured in order to maintain its optimal delivery. Unfortunately, users inadvertently share or expose their user credentials, which results to user account compromise, thus allowing other individuals take over their accounts. A potential solution is to apply the creation of user profiles (pattern detection) through

their web log activities and executed as a service over network that can reinforce and proactively execute the management for internet use and ultimately detecting possible compromised user account.

Profiling of user Web activity is an example implementation of Web Usage Data Mining. Web Usage Data Mining is the application of pattern mining techniques to usage logs of large Web data repositories in order to produce results that can be used in several implementation that focus on analytics and model creation (Gupta, 2006). In the profiling activity, its task is to generate unique user profiles/ classes sourced from browsing activity logs with reference to visited websites, IP addresses, user access log dates and timestamps. This will then be delivered as an application, where its main function is to compare present and future web use vs the created user profiles, and generating reports of either an ideal utilization or possible web use anomaly (eg. compromised user account).

In this paper, the author presents the development of a web application that will possibly detect anomaly in the user web activity by creating profiles using the Naïve Bayes classifier. First phase will be the preprocessing activity where it creates the corpus consisting of the web activity logs and user log-ins. Then in the Model generation, it goes through the training process, resulting to the different user profiles. In comparing with new user activities computed likelihood scores are generated.

2 Review of Related Work

Naïve Bayes provides a simple approach with clear semantics that returns impressive results in terms of classification activity (Witten, Frank, & Hall, 2011). In the study entitled Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text, Naïve Bayes is compared against Support Vector Machine (SVM) and C4.5 Classifier using the Weka (Weikato Environment for Knowledge Analysis) toolkit (Wahbeh & Al-Kabi, 2012). In terms of accuracy, the result showed that the Naïve Bayes outperformed the other two classifiers.

In the research entitled "*Identifying High-Level Student Behavior Using Sequence-based Motif Discovery",* its objective is to do a profiling of students that will reinforce pedagogy. It involves a data mining technique called "Sequence-based Motif Discovery", were it is being used for detecting student behavior patterns which is embedded as a function in the tutoring system developed which is utilized as an input to strategize the tutoring scheme. With the discovered motifs, this were used to classify the determined/ captured student behaviors, and then was utilized in reinforcing the tutoring/ teaching with respect to the capability of the student (Shanabrook, Cooper, Woolf, & Arroyo, 2010).

The paper of Johannes Furnkranz (Furnkranz, 1998), presents the use of n-gram specifically the bigram and trigram (length of 2 and 3), resulted to a more functional text categorization activity, and anything greater would result to a reduced classification performance. This is also supported in the research work of Andelka Zecevic (Zecevic , 2011), where the sizes of three (3) and four (4) yields good performance in authorship text classification.

3 Preprocessing Activity and the use of Naïve Bayes Classifier

Since this paper involves detecting user web activity patterns, this can be relative to a document classification or a multi-class exercise where prior to fitting the generated model and using the chosen machine learning algorithm for training, in this case using the Naïve Bayes Classifier, we use the N-Gram model, which is a variation of the bag of words approach to represent the recordset as a feature vector. In the preprocessing activity, prior to executing, the bag of words approach is utilized by seeing the document a set that contains all the words in the document with multiple occurrences, relatively, a set refers to its members just once, while a bag can have repeated elements (Witten, Frank, & Hall, 2011). In a given document classification exercise, in each occurrence, it represents a document and the occurrence‘s class as the document topic. Documents are characterized by the words, and each instance of the word, whether present or absent is regarded as a Boolean Attribute (Frank & Bouckaert). Specifically, in the bag of words process, during the training process, we construct and analyze the document or in this case the visited sites in terms of n-grams, which will be idiom agnostic and in trigram form (Zecevic , 2011).

Using the Naïve Bayes Classifier, we adopt the Multinomial model where it captures the frequency of words, not just their presence or absence in comparison with the Bernoulli model (Raschka, 2014).

$$P_(x_i|\omega_j)=\sum tf(x_i,d\in\omega_j)+\alpha\sum Nd\in\omega_j+\alpha\cdot V$$

Where

x_i	A word or token from a particular sample.
$\sum tf(x_i,d\in\omega_j)$	The sum of raw term frequencies of word x_i from all documents in the training sample that belong to class \ ω_j.
$\sum Nd\in\omega_j$	The sum of all term frequencies in the training dataset for class \ ω_j.
α	An additive smoothing parameter (α=1 for Laplace smoothing).
V	The size of the vocabulary (number of different words in the training set given a class/ profile).

The likelihood score can then be expressed as p(S | W) where S refers to the user and W as the collection of websites, which then can be computed from log (p(S | W) / p(¬S | W)) based on the observation that p(S | W) + p(¬S | W) = 1 (Shimodaira, 2015) (Collins, 2016).

4 Objectives

This paper aims to develop a web application that will detect web usage anomaly through User Profiling using Naïve Bayes Classifier. This will have two activities:

1) **Training Process** – using logs taken from the users' web activities vis-à-vis IP addresses, date and time stamp, a supervised classification data mining process will take place using the Naïve Bayes Classifier, creating the unique user profiles.
2) **Testing/ Detection Process** – using the user profiles created, the remaining user web logs (web activities, IP addresses, date and time stamp) are loaded and compared to the generated user profiles in the knowledgebase determining its similarity or likelihood.

5 Methodology

Figure 1, presents the conceptual diagram of the paper, where it starts off with the data preparation and transformation. The two repositories namely the web logs which contains the IP address or the site visits, local machine IP address and the user activity logs which contains users and the local machine IP address are being merged into one recordset. In both repositories, the common attributes are the date/ time stamp and the machine or the local IP address where the user has logged in, which is used as basis in merging.

The resulting merge recordset will have the following attributes, unique users, web site visited, date and time stamp and the local machine address, this is stored in a table named tbl*report* in MySQL database. The training data set, where it covers three (3) months from April 2015 to July 2015, and for the testing set, covers the month of July 2015, making the distribution 75% and 25% respectively. In the training process, which is the primal activity of the model building, will have a table containing the training set. The required iteration will be dependent with the number of user occurrence. For every user occurrence, the associated records (web site visited) is then analyzed using the Naïve Bayes classifier, which then generates the corresponding user profile with its associated N-grams (Al-diabat, 2012) (Kešelj, Peng, Cercone, & Thomas, 2003), subsequently, storing it in the tbl*knowledgebase* table.

In the testing phase, where we are applying already user profile or the knowledge models created, the corresponding testing set is loaded and compared to the generated user profile. A corresponding likelihood score is computed. In the development of this paper, core PHP is used and Apache for web service delivery.

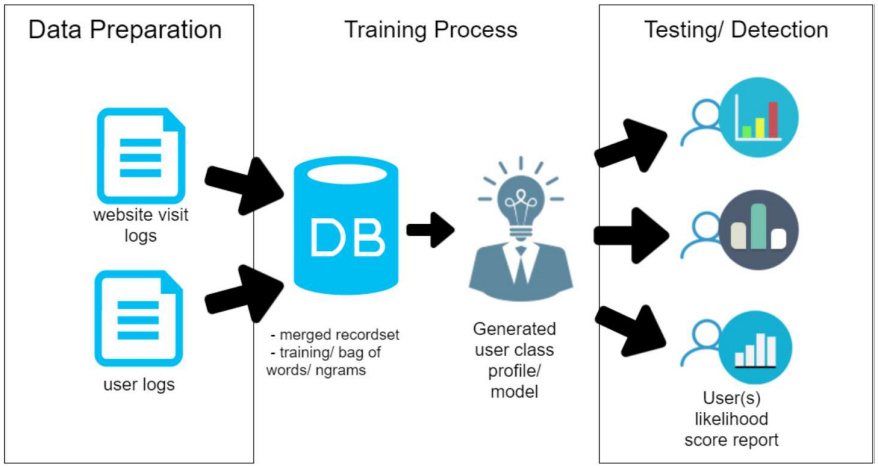

Figure 1. Conceptual Diagram

6 Development Process

6. a. Training Activity

In the flowchart presented in Figure 2 which is in the succeeding page, a table named *tbltraining* contains the training data set extracted from the table *tblreport*, which is the consolidation of the two recordsets coming from these sources, namely the wifi log-ins and the visited sites logs. In going through the training,, a class named *trainer()* handles the execution, where in each activity, the training process is filtered or classified by the user account. The table *tblexample*, which contains the users and all its associated sites visited as extracted from *tbltraining*, serves as the repository for the pattern extraction function, named *extractPatterns()*.

In this function, we implement the N-gram method, where we parse through the visited sites and represent them in a trigram order. The end result of this training, is that we are able to contain all the user profiles, with their associated N-grams and the number of occurrence. The table *tblknowledgebase* contains these information.

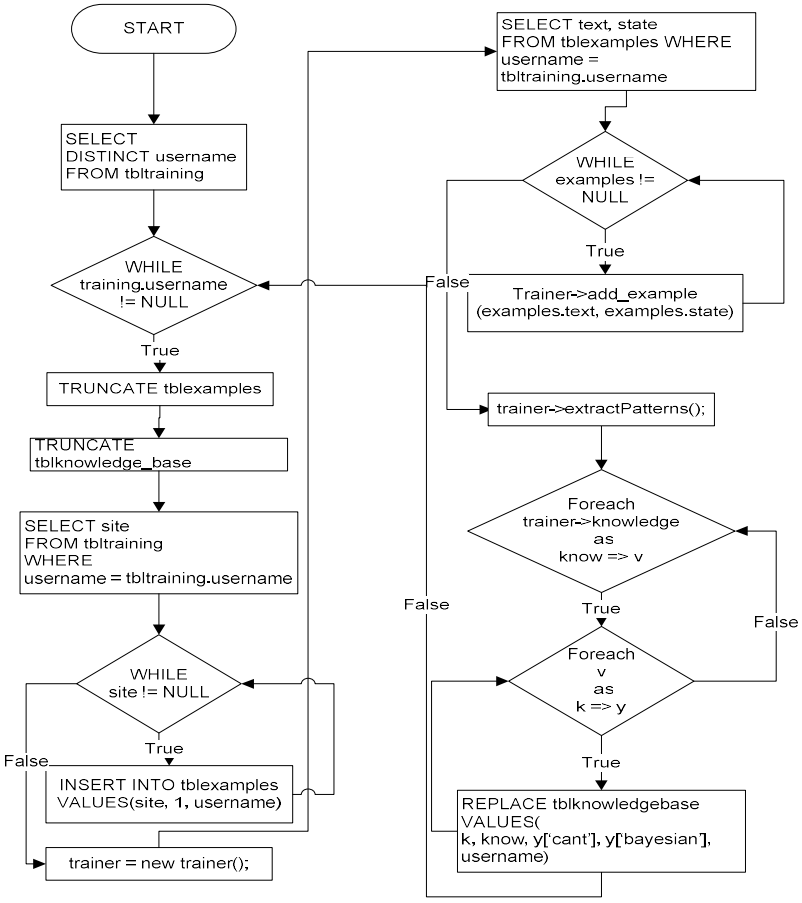

Figure 2. Training Flowchart

6.b. Testing/ Detection Activity

The testing/detection activity shown in Figure 3, starts with the loading of the remaining data set, which is the July 2015 web activities. As discussed previously, for the training set used, it covers from April 2015 to June 2015, which is three (3) months and for the testing set covers the month of July 2015, making the distribution 75% and 25% respectively. With the identification of the testing set, this will then be compared with the generated profiles. In order to have a better appreciation of the result, the computed likelihood score together with the corresponding user profile are stored in a table named *tblresult*. An iteration is being executed to retrieve the generated user profiles along with their generated N-grams from the *tblknowledgebase*.

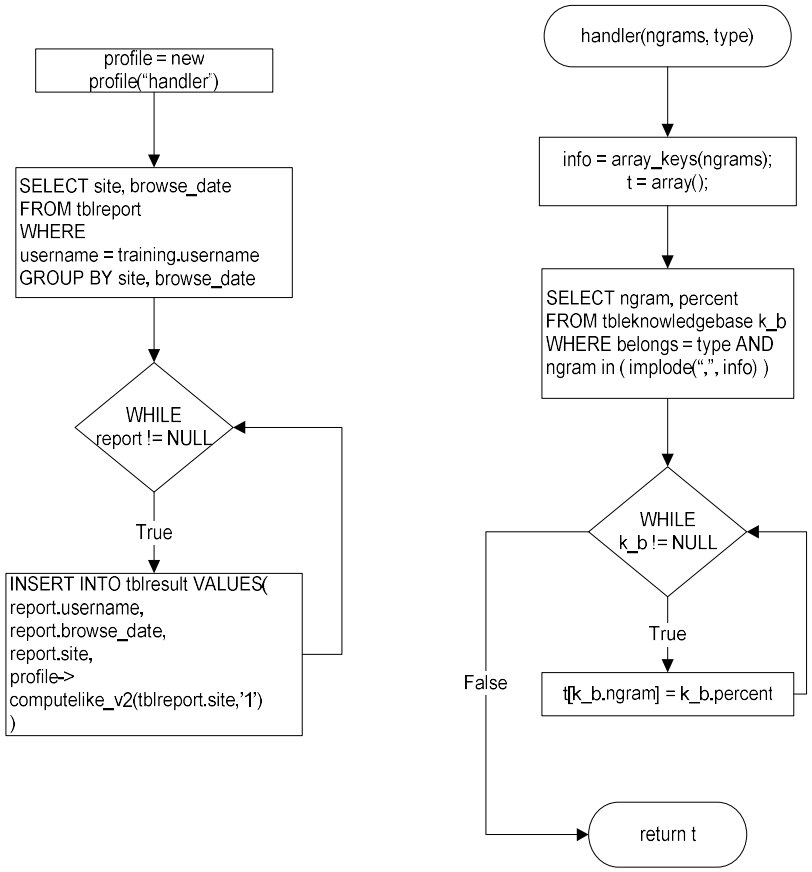

Figure 3. Testing/ Detection Activity

In Figure 4 below, provides the code snippet for computing the likelihood score. Simply, for each website visited, we look at each of the generated n-gram given the associated user profile. The *ngrams* variable holds the number of times we've seen the n-gram used during the training and the *knowledge* variable contains the product of all n-grams that are extracted from the table *tblknowledgebase*. In computing the likelihood score, to avoid the possibility of encountering the "floating point underflow" error, where the computed numbers would get too small to be handled properly, addressing this, we compute using the log function of the prior and current values, H and S variables respectively. Further the use of chi-square, also added to the fitness of the computed likelihood score (Pinheiro & Bates, 1995) (Rodas, 2008).

```
function computelike_v2($text,$type) {
    $ngram = new ngram;
    $ngram->setText($text);

    for($i=3; $i <= 5;$i++) {
        $ngram->setLength($i);
        $ngram->extract();
    }

    $fnc = $this->_source;
    $ngrams =  $ngram->getnGrams();
    $knowledge =  $fnc( $ngrams,$type );
    $total=0;
    $acc=0;
    $N = 0;
    $H = $S = 1;

    foreach($ngrams as $k => $v) {
        if ( !isset($knowledge[$k]) ) continue;
        $N++;
        $value = $knowledge[$k] * $v;
        $H *= $value;
        $S *= (float)( 1 - ( ($value>=1) ? 0.99 : $value) );
    }

    $H = $this->chi2Q( -2 * log( $N *  $H), 2 * $N);
    $S = (float)$this->chi2Q( -2 * log( $N *  $S), 2 * $N);
    $percent = (( 1 + $H - $S ) / 2) * 100;
    return is_finite($percent) ? $percent : 100;
}
```

Figure 4: Code Snippet for Likelihood Score Computation

7. TESTING AND OBSERVATIONS

The training activity has generated 3690 unique user profiles or models. The result of the testing activity which involved 25% of the available recordset, specifically the July, 2015 data as the test set, has generated a total of 615,056 rows. For the purpose of

validating the testing results we have identified a sample size of 384 test cases (Computing the Sample Size, 2016). Noting the computed likelihood scores, manually reviewing the web site visits in the test data, we have observed that any value lesser than 5% tends to direct to a site which is not previously visited and any computed score which is at least 85% relates to a previously visited website.

Table 1. Confusion Result Matrix

N=384	Predicted: No	Predicted: Yes
Actual: No	11	4
Actual: Yes	22	347

N = refers to the number of test activities

In the Confusion Matrix Result (Confusion Matrix, 2014) presented in Table 1, the test shows that accuracy is 93% where it predicted correctly 11 instances of none aligned visits and 347 sites correctly identified as previously visited. Further, out of 369 expected previous web site visits, 22 were not predicted properly. 17 of these are numeric url's and the likelihood score is considered insignificant, having a consistent computed likelihood score of 76%, which is consistent also with those sites which are expected to be not previously visited,.

8 CONCLUSION AND FUTURE UNDERTAKINGS

The core objective of this paper is to generate a report per user of a likelihood score referring to the generated user profile vis-à-vis his current browsing activity. This simply helps the administrator qualify for further action in determining whether an account is compromised or not, or the need to further review or observe. Further, building a user class/ profile using naïve bayes and then computing the likelihood score provides a rather strong conclusion whether or not a given web activity vis-à-vis the user is a valid/ associated web activity to the user.

Using the naïve bayes classification, integrating and using it in the prototype development, provides a simpler process especially in conjunction with the use of the bag of words specifically the N-grams approach. A possible future activity to strengthen the detection process will be the integration of sequential pattern mining strategies.

9 REFERENCES

Collins, M. (2016, August). *The Naive Bayes Model, Maximum-Likelihood Estimation and the EM Algorithm.*

Al-diabat, M. (2012). Arabic Text Categorization Using Classification Rule Mining. *Applied Mathematical Sciences, 6*(81), 4033 - 4046.

Computing the Sample Size. (2016, August 8). Retrieved from Creative Research Systems: http://www.surveysystem.com/sscalc.htm

Confusion Matrix. (2014, March 26). Retrieved from Data School of Machine Learning: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Frank, E., & Bouckaert, R. R. (n.d.). Naive Bayes for Text Classification with Unbalanced Classes.

Furnkranz, J. (1998). A Study Using n-gram Features for Text Categorization.

Gupta, G. (2006). *Introduction to Data Mining with Case Studies.* Prentice-Hall Of India Pvt. Limited.

Kešelj, V., Peng, F., Cercone, N., & Thomas, C. (2003). N-gram-based author profiles for authorship attribution. *Pacific Association for Computational Linguistics, PACLING*, (pp. Volume 3, 255–264).

Oracle. (2008). *Oracle Data Mining Concepts, 11g*. Oracle.

Pinheiro , J. C., & Bates, D. M. (1995). Approximations to the Loglikelihood Function in the Nonlinear Mixed Effects Model. *Journal of Computational and Graphical Statistics*.

Raschka, S. (2014, October 4). *Naive Bayes and Text Classification*. Retrieved from http://sebastianraschka.com/Articles/2014_naive_bayes_1.html

Rodas, C. D. (2008). *Bayesian Spam Filter: Detect spam in text using Bayesian techniques*. Retrieved from phpclasses.org: http://www.phpclasses.org/package/4236-PHP-Detect-spam-in-text-using-Bayesian-techniques.html

Shanabrook, D. H., Cooper, D. G., Woolf, B., & Arroyo, I. (2010). Identifying High-Level Student Behavior Using Sequence-based Motif Discovery. *Proceedings of the 3rd International Conference in Data Mining*.

Shimodaira, H. (2015). Text Classification using Naive Bayes.

Wahbeh, A. H., & Al-Kabi, M. (2012). Comparative Assessment of the Performance of Three. *ABHATH AL-YARMOUK: "Basic Sci. & Eng."*.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition.* MORGAN KAUFMANN PUBLISHERS.

Zecevic , A. (2011). N-gram Based Text Classification According To Authorship.