Distributed Phishing Detection by Applying Variable Selection using Bayesian Additive Regression Trees

Saeed Abu-Nimeh[1], Dario Nappa[2], Xinlei Wang[2], and Suku Nair[1]
SMU HACNet Lab
Computer Science and Engineering Dept.
Southern Methodist University
Dallas, TX 75275
[1]{sabunime, nair}@engr.smu.edu
[2]{dnappa, swang}@smu.edu

***Abstract*—Phishing continue to be one of the most drastic attacks causing both financial institutions and customers huge monetary losses. Nowadays mobile devices are widely used to access the Internet and therefore access financial and confidential data. However, unlike PCs and wired devices, such devices lack basic defensive applications to protect against various types of attacks. In consequence, phishing has evolved to target mobile users in Vishing and SMishing attacks recently. This study presents a client-server distributed architecture to detect phishing e-mails by taking advantage of automatic variable selection in Bayesian Additive Regression Trees (BART). When combined with other classifiers, BART improves their predictive accuracy. Further the overall architecture proves to leverage well in resource constrained environments.**

I. Introduction

Nowadays phishing attacks appear in various types and forms. Yet, traditional attacks delivered by spoofed e-mails remain the dominant type of phishing. Here the bad actor forges e-mails falsely mimicking legitimate ones and thus mails them to victims using *mailers*. Victims are then lured into divulging their confidential credentials, such as credit card information, social security numbers, or online login credentials. *Vishing*, or Voice over Internet Protocol (VoIP) phishing, has recently emerged as a new vector of phishing attacks, as it is easy to setup and take down by phishers. The attack can be carried by setting up a free VoIP account then using caller ID spoofing to mimic legitimate financial institutions' phone numbers.

Furthermore, because of the ubiquity of mobile devices and the various applications to access the Internet therein, many users are using blackberries, PDAs, or even cell phones to access their bank accounts and store sensitive personal data. New forms of phishing attacks that target mobile devices are on the rise. SMS phishing, dubbed as *SMishing*, is an emerging vector of phishing attacks where the victim receives a short message service (SMS) and thus is lured into clicking on a URL to download malware or is redirected to fraudulent sites.

Surly, there are merely few solutions available to mitigate phishing attacks in mobile devices. In addition, several ubiquitous solutions available for desktop and wired computers are generally not as readily available across wireless and mobile devices. This is due to several known limitations in such devices. Due to power constraints, processing capabilities and storage capacities are limited, which in return affect security and privacy solutions built for such devices to protect users against various attacks. As a result, various attacks, including phishing, can easily take advantage of the limited or lack of security and defense applications in these devices.

Although Bayesian Additive Regression Trees (BART) has proven to be competitive in classifying spam e-mails, previous research [1] showed that it is very demanding in terms of memory consumption and learning computational time. In consequence, it cannot be deployed in resource constrained devices. In this study we propose a distributed architecture for the detection of phishing e-mails in a mobile environment. The motivation behind the distributed architecture is to harden the attack detection at the client level and conceal the overhead associated with BART at the server level. A mutual feedback mechanism is deployed between the server and the clients. At the server side, that is the MTA (mail transfer agent), BART is applied to classify the majority of the e-mails received by the MTA. At the client side, *lighter* machine learning approaches are used to classify phishing e-mails in resource constrained devices taking advantage of automatic variable selection in BART.

The rest of the paper is organized as follows. In Section II we present related work and describe BART briefly. In Section III we explain our distributed architecture in details. Section IV demonstrates the experimental studies. The results are discussed in Section V. We draw conclusion and motivate for future work in Section VI.

II. Related Work

In [2], the authors investigated the application of Hill Climbing, Simulated Annealing, and Threshold Accepting techniques as feature selection algorithms for spam filtering and compared their performance against Linear Discriminate

Analysis. The results demonstrated that these techniques can be used not only to reduce the dimensions of the e-mail, but also improve the performance of the classification filter. In addition, there exist several approaches that measure the importance and the effectiveness of a certain feature in the overall prediction process. Such algorithms are known as feature ranking approaches [3]. In [4], the authors applied simulated annealing as an algorithm for feature selection on a phishing dataset. After a feature set was chosen, they used information gain (IG) to rank these features based on their relevance.

In addition, Chandrasekaran et al. [4] proposed a technique to classify phishing based on structural properties of phishing e-mails. They applied one-class SVM to classify phishing e-mails based on selected features. Their results claim a detection rate of 95% of phishing e-mails with a low false positive rate. Fette et al. [5] compared a number of commonly-used learning methods through their performance in phishing detection on a past phishing dataset, and finally Random Forests were implemented in their algorithm PILFER. The proposed method detected correctly 96% of the phishing e-mails with a false positive rate of 0.1%. Abu-Nimeh et al. [6] compared six machine learning techniques to classify phishing e-mails. They showed that, by merely using a *bag-of-words* approach, the studied classifiers could successfully predict more than 92% of the phishing e-mails.

In the approach we propose here we neither worry about variable selection, nor feature ranking techniques, as BART supports automatic variable selection. Thus, BART chooses the best variables that represent the relationship between the features and the response in the dataset during the training phase. We describe BART in more details in the following section.

A. Bayesian Additive Regression Trees

Bayesian Additive Regression Trees (BART) is a new learning technique, proposed by [7], to discover the unknown relationship between a continuous output and a dimensional vector of inputs. The original model of BART was designed for regression problems; however, in [1] the authors modified it (and named it CBART) to be applicable to classification.

BART discovers the unknown relationship f between a continuous output Y and a p dimensional vector of inputs $x = (x_1, ..., x_p)$. Assume $Y = f(x)+\epsilon$, where $\epsilon \sim N(0, \sigma^2)$ is the random error. Motivated by ensemble methods in general, and boosting algorithms in particular, the basic idea of BART is to model or at least approximate $f(x)$ by a sum of regression trees,

$$f(x) = \sum_{i=1}^{m} g_i(x); \qquad (1)$$

each g_i denotes a binary tree with arbitrary structure, and contributes a small amount to the overall model as a *weak learner*, when m is chosen large. Each of the interior (i.e., non-terminal) nodes in the binary tree is associated with a binary splitting rule based on some x variable. By moving downwards from the root, an observation with given x will be assigned to a unique terminal node, according to the splitting rules associated with the nodes included in its path. In consequence, the corresponding parameter of the terminal node will be the value of g for this observation.

Let T_i be the i^{th} binary tree in the model (1), consisting of a set of decision rules (associated with its interior nodes) and a set of terminal nodes, for $i = 1, \cdots, m$. Let M_i be the vector containing all terminal node parameters of T_i such that $M = \{M_1, \cdots, M_{b_i}\}$ and b_i is the number of terminal nodes that T_i has. Now we can explicitly write

$$Y = g(x; T_1, M_1) + \ldots + g(x; T_m, M_m) + \epsilon. \qquad (2)$$

Figure 1 depicts an example of a binary tree in the BART model. Note that the BART contains multiple binary trees, since it is an additive model. Each node in the tree represents a feature in the dataset and the terminal nodes represent the probability that a specific e-mail is phishing, given that it contains certain features. For example, if an e-mail contains HTML code, contains javascript, and the javascript contains form validation, then the probability that this e-mail is phishing is 80% (according to the example in Figure 1).

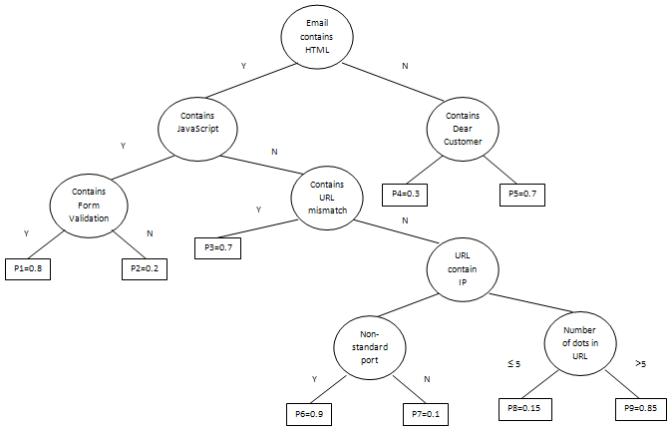

Fig. 1. Example of a binary tree.

BART is fully model-based and Bayesian in the sense that a *prior* is specified, a *likelihood* is defined using the data, and then a sequence of draws from the *posterior* using Markov chain Monte Carlo (MCMC) is obtained. Specifically, a *prior* distribution is needed for T, M, and σ, respectively. Each draw represents a fitted model f^* of the form (1). Due to space constrains, we do not provide in depth discussion of BART. Though the interested reader can refer to [1], [7] and the references therein for further information.

BART has several features that render it competitive to other well-known classifiers. BART automatically selects variables from a large pool of input predictors, while searching for models with highest posterior probabilities for future prediction, via a backfitting MCMC algorithm. Compared to other Bayesian methods, such as Naive Bayes and Bayesian Networks, the latter approaches require variable selection to be

done separately, otherwise they use all the variables supplied for training, thus the performance of the classifier will be very poor. In addition, it is well known that variable selection in a high dimensional space is a very difficult problem that often requires intensive computations. Note that phishing e-mails change regularly and vastly to lure detection mechanisms and the phishing features in e-mails may change over time as well. Yet, the above nice feature of BART comes handy when training on newly arriving e-mails on a regular basis. With no additional requirements to perform variable selection, BART simultaneously accomplishes variable selection during the training phase.

In addition, in phishing detection hundreds of potential features are extracted from raw e-mails. Only an unknown subset of them is useful for prediction, however others may be deemed irrelevant. Consequently, blindly including all the variables during training often leads to overfitting, and hence predicting new attacks may be poor. However, with the automatic variable selection feature in BART this problem is solved.

III. Distributed Phishing Detection

Based on the results of previous research [1], CBART outperforms other classifiers when predicting spam e-mails. Yet, the overhead associated with CBART renders its implementation impractical in resource constrained devices due to the limitations discussed earlier. Albeit, the implementation of CBART may be suitable in a server environment due to the abundance of resources therein (i.e. processing, power, and memory).

Our main goal here is to take advantage of the superior predictive accuracy of CBART to detect the majority of phishing e-mails at the server level. Further, by deploying CBART at the server level, the overhead associated with CBART can be concealed. Afterwards automatic variable selection in CBART can be used to improve the predictive accuracy in client devices. After CBART performs variable selection automatically and generates the sum-of-tree model, the selected variables are fed to clients, so classifiers in clients can use them when predicting classes of new e-mails. By doing this, the predictive accuracy of clients is expected to improve, since merely the features of interest are used during classification. Performing variable selection at the server level not only improves the predictive accuracy of clients, but also eliminates extra computational time and processing overhead in clients needed to do so. Figure 2 depicts a high level description of the architecture.

IV. Experiential Studies

A. Phishing Dataset

A phishing dataset is constructed from 6561 raw e-mails. 1409 among these e-mails are phishing donated by [8] covering many of the new trends in phishing and collected between August 7, 2006 and August 7, 2007. The legitimate portion of the dataset is 5152 e-mails, which are collected from financial-related and other regular communication e-mails.

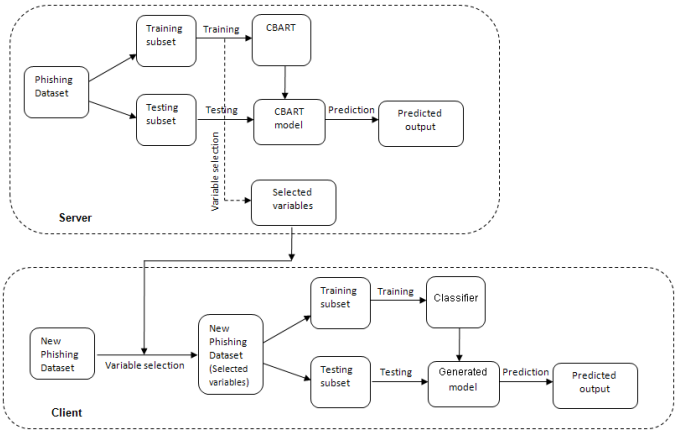

Fig. 2. Distributed phishing detection using variable selection block diagram.

The financial-related e-mails are sent by financial institutions such as Bank of America, eBay, PayPal, American Express, Chase, Amazon, AT&T, and many others. The remaining part of the legitimate set is collected from the authors' mailboxes. These e-mails represent regular communications, e-mails about conferences and academic events, and e-mails from several mailing lists. Table I summarizes the ratio of the e-mails in the dataset.

TABLE I
Corpus description.

Corpus	No. of e-mails	Percentage (%)
Phishing	1409	21%
Legitimate (financial)	178	3%
Legitimate (other)	4974	76%
Total	6561	100%

The dataset constitutes of 71 features, in which the first feature represent the class of the e-mail, whether it is phishing =1 or legitimate =0. Thus, the following 60 features represent the terms that frequently appear in phishing e-mails gauged by term frequency inverse document frequency (TF/ IDF). TF/IDF calculates the number of times a word appears in a document multiplied by a (monotone) function of the inverse of the number of documents in which the word appears. Therefore, terms that appear often in a document and do not appear in many documents have a higher weight [9]. The last 10 features represent structural characteristics of phishing e-mails and several styles used by phishers to lure victims and make phishing e-mails look legitimate.

B. Experimental Setup

We compare the classifiers' performance using multiple measures. Primarily, we use the area under the receiver operating characteristic (ROC) curve (AUC) to evaluate the performance of classifiers. The AUC shows the trade off between the false positives and true positives at different cut-off points. In addition, we use classifiers' error rate (Err), false positive rate (FP), and false negative rate (FN) to gauge the performance of classifiers. Although classifiers' error rate

(Err) or classifiers' accuracy (Acc), that is $1 - Err$, have been widely used in comparing classifiers' performance, they have been criticized for highly depending on the probability of the threshold chosen to approximate the positive classes. Therefore, some researchers recommend using AUC than classifiers' error rate or accuracy when evaluating classifiers' performance [10]. Here we note that, when using the error rate, we assign new classes to the positive class if the probability of the class is greater than or equal to 0.5 (i.e. threshold=0.5).

False positive rate is the total number of legitimate e-mails misclassified as phishing ($n_{L \to P}$) divided by the total number of legitimate e-mails (N_L).

$$FP = \frac{n_{L \to P}}{N_L}. \tag{3}$$

False negative rate is the total number of phishing e-mails misclassified as legitimate ($n_{P \to L}$) divided by the total number of phishing e-mails (N_P).

$$FN = \frac{n_{P \to L}}{N_P}. \tag{4}$$

The error rate is the total number of misclassified e-mails divided by the total number of e-mails (legitimate and phishing) in the dataset.

$$Err = \frac{n_{L \to P} + n_{P \to L}}{N_L + N_P}. \tag{5}$$

We optimize the classifiers' performance by testing them using different input parameters, as shown in Table II. In order to find the maximum AUC, we test the classifiers using the complete dataset applying different input parameters. In addition, to find the minimum average error rate we apply *10-fold-cross-validation* and average the estimates of all 10 folds (sub-samples). Note that when calculating the AUC values, the classifiers are tested using the complete dataset without applying 10-fold-cross-validation.

Variable selection is performed using automatic variable selection in CBART. CBART selects the variables that are frequently used during MCMC simulations. In our experiments the 6 most frequently used variables are selected. Note that we performed our experiments using different number of selected variables, namely 10, 20, and 30, though 6 variables achieved the maximum AUC. To evaluate the effectiveness of variable selection in CBART, we test it against Kruskal-Wallis (KW) test [11]. KW test can be used to measure the importance of predictors based on their p-value. In our experiments we select all variables with $p - value < 0.1$. Thus the total number of selected variables based on the KW test is 13 variables.

C. Experimental Results

In this section we present the experimental results by comparing the AUC, FP, FN, and Err using the optimum parameters achieved in the previous section.

Table III and Table IV compare the AUC, error rate, false positive, and false negative rates before and after applying the distributed approach respectively. Here we compare the performance of classifiers before using variable selection and after selecting 6 variables by CBART. Table V summarizes the improvement or decay in classifiers after applying variable selection. Note that the performance of classifiers degrades when the AUC decreases and/or the error rate, FP, FN increases.

Table VI compares the AUC, error rate, false positive, and false negative rates of classifiers after applying variable selection using the top 13 variables in KW test. Table VII summarizes the improvement or decay in classifiers after applying variable selection.

TABLE II
OPTIMIZED INPUT PARAMETERS IN CLASSIFIERS.

Classifier	Input parameters	
CBART	number of trees= 100	$power = 1$
LR	$\lambda = 1 \times 10^{-4}$	
RF	number of trees= 50	
SVM	$\gamma = 0.1$	cost $(c) = 12$
NNet	size $(s) = 35$	weight decay $(w) = 0.7$

TABLE III
AUC, ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES BEFORE APPLYING THE DISTRIBUTED APPROACH.

Classifier	AUC	Err	FP	FN
CART	96.06%	7.00%	11.55%	22.10%
RF	95.48%	3.68%	4.25%	13.20%
SVM	97.18%	2.39%	5.43%	13.77%
LR	54.45%	5.34%	7.29%	18.38%
NNet	98.80%	4.31%	6.16%	14.32%

TABLE IV
AUC, ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES AFTER APPLYING THE DISTRIBUTED APPROACH. SIX VARIABLES ARE SELECTED USING AUTOMATIC VARIABLE SELECTION IN CBART.

Classifier	AUC	Err	FP	FN
CART	94.49%	7.09%	11.31%	22.81%
RF	93.60%	2.85%	2.60%	10.90%
SVM	95.01%	5.37%	8.79%	16.93%
LR	62.85%	3.37%	4.14%	11.83%
NNet	94.95%	3.27%	3.77%	11.74%

TABLE V
INCREASE OR DECREASE IN AUC, ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES AFTER APPLYING THE DISTRIBUTED APPROACH USING AUTOMATIC VARIABLE SELECTION IN CBART. THE HIGHER THE AUC VALUE THE BETTER THE CLASSIFIER'S PERFORMANCE. THE LOWER THE ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES, THE BETTER THE CLASSIFIER'S PERFORMANCE.

Classifier	AUC	Err	FP	FN
CART	-1.57%	+0.09%	-0.24%	+0.71%
RF	-1.88%	-0.83%	-1.65%	-2.30%
SVM	-2.17%	+2.98%	+3.36%	+3.16%
LR	+8.40%	-1.97%	-3.15%	-6.55%
NNet	-3.85%	-1.04%	-2.39%	-2.58%

V. DISCUSSION

The present study proposes a client-server architecture to detect phishing e-mails in a resource constrained environment.

TABLE VI
AUC, ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES AFTER APPLYING THE DISTRIBUTED APPROACH. 13 VARIABLES ARE SELECTED USING KW TEST.

Classifier	AUC	Err	FP	FN
CART	94.90%	7.97%	15.55%	22.71%
RF	94.57%	4.60%	6.21%	15.72%
SVM	97.09%	5.58%	7.91%	18.90%
LR	37.15%	7.44%	8.58%	27.73%
NNet	96.36%	4.97%	7.53%	16.22%

TABLE VII
INCREASE OR DECREASE IN AUC, ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES AFTER APPLYING THE DISTRIBUTED APPROACH USING KW TEST. THE HIGHER THE AUC VALUE THE BETTER THE CLASSIFIER'S PERFORMANCE. THE LOWER THE ERROR RATE, FALSE POSITIVE, AND FALSE NEGATIVE RATES, THE BETTER THE CLASSIFIER'S PERFORMANCE.

Classifier	AUC	Err	FP	FN
CART	-1.16%	+0.97%	+4.00%	+0.61%
RF	-0.91%	+0.92%	+1.96%	+2.52%
SVM	-0.09%	+3.19%	+2.48%	+5.13%
LR	-17.30%	+2.10%	+1.29%	+9.35%
NNet	-2.44%	+0.66%	+1.37%	+1.90%

CBART is deployed at the server to detect the majority of phishing e-mails. Thus the associated clients use the variables selected by CBART during training to improve their predictive accuracy and eliminate the overhead needed to perform variable selection. In addition, the effectiveness of CBART's variable selection is compared against another variable selection approach, namely Kruskal-Wallis (KW) test. The results demonstrate that when using variable selection via CBART, the AUC of all classifiers decreases, except in LR, indicating a decay in classifiers' performance. Similarly, when using variable selection via KW the AUC in all classifiers decreases. On the other hand, when using variable selection via CBART, the error rate, false positive rate, and false negative rate decrease, in all classifiers, except in SVM, showing an improvement in classifiers' performance. One of the known disadvantages of SVM is that it is prone to overfitting. We expect that this is the reason behind the decay in the performance of SVM. When SVM is trained on more variables, it overfits the data and the performance of classifiers improves (see Table III). However, when the number of variables decreases noticeably, the performance of SVM decreases as shown in Table IV.

To the contrary, when using a different variable selection approach, namely KW test, the selected variables do not improve the performance in any of the classifiers (see Table VII). The AUC in all classifiers decreases as we mentioned earlier, indicating a decay in performance, and the error rate, false positive rate, and false negative rate increase, showing a decay in performance as well.

We believe that the overhead associated with CBART can be concealed if applied in a server environment. Further, other classifiers can benefit from automatic variable selection in CBART as demonstrated in the experiments. Moreover, the automatic variable selection in CBART proves to be competitive to at least variable selection in KW. In summary, the proposed distributed architecture looks promising and suitable for phishing detection in a resource constrained environments.

VI. CONCLUSIONS AND FUTURE WORK

Phishers are exploiting new attack vectors to lure mobile users. Several ubiquitous solutions available for desktop and wired computers are generally not as readily available across wireless and mobile devices due to processing, power, and storage limitations. The present study proposed a distributed client-server architecture to detect phishing attacks in a mobile environment. CBART was implemented at the server to detect the majority of phishing e-mails. Thus associated clients took advantage of automatic variable selection in CBART to improve their predictive accuracy and eliminate the overhead of variable selection is applied.

The results demonstrated that automatic variable selection in CBART can be used to improve the predictive accuracy in other classifiers. Although the AUC decreased for the majority of classifiers (except LR), the error rate, false positive rate, and false negative rate decreased for RF, LR, and NNet after using variable selection via CBART. However, when using another variable selection technique, namely Kruskal-Wallis (KW) test, the predictive accuracy for all the compared classifiers degraded.

The results motivate future work to compare the effectiveness of automatic variable selection in CBART against other well-known variable selection approaches to derive more extensive conclusions.

REFERENCES

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "Bayesian additive regression trees-based spam detection for enhanced email privacy," in *ARES '08: Proceedings of the 3rd International Conference on Availability, Reliability and Security*, 2008, pp. 1044–1051.

[2] R. Wang, A. M. Youssef, and A. K. Elhakeem, "On some feature selection strategies for spam filter design," in *CCECE '06: Canadian Conference on Electrical and Computer Engineering*, 2006, pp. 2186–2189.

[3] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=944968

[4] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *NYS Cyber Security Conference*, 2006.

[5] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2007, pp. 649–656.

[6] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. New York, NY, USA: ACM, 2007, pp. 60–69.

[7] H. A. Chipman, E. I. George, and R. E. McCulloch, "BART: Bayesian Additive Regression Trees," 2006, http://faculty.chicagogsb.edu/robert.mcculloch/research/code/BART-7-05.pdf.

[8] J. Nazario. (2007) Phishing Corpus. Http://monkey.org/ jose/phishing/phishing3.mbox.

[9] M. W. Berry, Ed., *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer, 2004.

[10] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, 2005.

[11] A. L. C. Howard B. Lee, *Elementary Statistics: A Problem Solving Approach*, 4th ed. Lulu.com, 2006.