International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:2, 2014

Incorporating Multiple Supervised Learning Algorithms for Effective Intrusion Detection

Umar Albalawi, Sang C. Suh, Jinoh Kim

Abstract**—**As internet continues to expand its usage with an enormous number of applications, cyber-threats have significantly increased accordingly. Thus, accurate detection of malicious traffic in a timely manner is a critical concern in today's Internet for security. One approach for intrusion detection is to use Machine Learning (ML) techniques. Several methods based on ML algorithms have been introduced over the past years, but they are largely limited in terms of detection accuracy and/or time and space complexity to run. In this work, we present a novel method for intrusion detection that incorporates a set of supervised learning algorithms. The proposed technique provides high accuracy and outperforms existing techniques that simply utilizes a single learning method. In addition, our technique relies on partial flow information (rather than full information) for detection, and thus, it is light-weight and desirable for online operations with the property of early identification. With the mid-Atlantic CCDC intrusion dataset publicly available, we show that our proposed technique yields a high degree of detection rate over 99% with a very low false alarm rate (0.4%).

Keywords**—**Intrusion Detection, Supervised Learning, Traffic Classification.

I. Introduction

THE number of Internet applications has significantly increased over the past decade with explosive usage of portable devices such as smartphones and tablets as well as personal computers. At the same time, there have been increasing demands on traffic classification to identify applications for various purposes, such as traffic engineering, resource provisioning, network usage and statistics, and security [1]–[4]. As a result, accurately identifying and categorizing network traffic has become a critical concern in the research community. From the security perspective, such a traffic classification is also crucial to detect unwanted traffic, so as to protect computing resources, to save bandwidth, and to preserve legitimate traffic. In this work, we approach the problem of traffic classification for identifying malicious network flows.

In fact, traffic classification is a core element for Intrusion Detection Systems (IDSs) [5] that is used as a second line of defense to observe suspicious actions on the network. The IDS detects attacks by inspecting networking traffic, and the general techniques widely used for IDSs are signature-based [6] and anomaly-based [7]. The signature-based technique uses a set of well-known signatures collected from malicious traffic flows, and thus, it is limited to *known* attacks previously studied. On the other hand, the anomaly-based technique relies on profiles that specify normal and abnormal behaviors [8], and the IDS could detect malicious traffic by referring to the profile databases. A benefit of the anomaly-based technique would be the capability of detecting new and variant attacks since it does not rely on previously known patterns. Our approach in this work is closer to anomaly-based, and we use a set of ML techniques to identify harmful traffic from the network with a high degree of accuracy.

In addition to *accuracy* for detection, another critical requirement is the *timeliness* and the process should be done as early as possible in the beginning stage of the traffic flow. Such an *early detection* has many benefits. For instance, it could reduce space and time complexity only by referring to the first *N* packets in the network flow. In addition, it would be possible to notice suspicious activities to the administrator more quickly, so that he/she could react against activities as early as possible. Unlike offline classification where all discrimination flow information is available, online classification should rely on partial information of the flow in the classification process [9]. In this work, we also consider the requirement for online classification as well as accuracy.

The key contributions of this paper can be summarized as follows. We present a novel method for intrusion detection that incorporates a set of supervised learning algorithms. The proposed technique provides high accuracy and outperforms existing techniques that simply relies on a single learning algorithm. In addition, our technique relies on partial flow information (rather than full information) for detection, and thus, it is light-weight and desirable for online operations. We also present experimental results with public intrusion data sets that show our proposed method yields a high degree of accuracy over 99% with very low false alarm rate (0.4%).

The paper is organized as follows. In Section II, we introduce existing efforts closely related to our work. Section III will provide the information about intrusion data sets we used and flow features used for supervised learning. In that section, we also discuss feature selection for optimization. In Section IV, we present our detection method that incorporates multiple ML

Algorithms and report evaluation results in Section V. We finally conclude our presentation in Section VI.

We use the general definition of network flow based on five tuples of source IP address and port number, destination IP address and port number, and the transport layer protocol.

Umar Albalawi is research assistant at Computer Science Department, Texas A&M University, Commerce, TX 75428 USA (phone: 412-953-2588; e-mail: ualbalwi@ut.edu.sa).

Sang C. Suh is Professor and Head of Computer Science Department, Texas A&M University, Commerce, TX 75428 USA (e-mail: Sang.Suh@tamuc.edu).

Jinoh Kim is Assistant Professor at Computer Science Department, Texas A&M University, Commerce, TX 75428 USA (e-mail: Jinoh.Kim@tamuc.edu).

II. Related Work

Numerous researches have been conducted to classify network traffic. Port-based classification is the traditional approach that relies on transport-layer port numbers. The traditional approach has several limitations, as many applications randomize their ports and cannot detect new application [10]. In the meantime, some studies rely on payload information by deep packet inspection techniques [11]. However, those techniques have several limitations. First, they fail to classify encrypted traffic. Second, there could be privacy concerns with examining user data. Another alternative is techniques that rely on flow features with ML algorithms, for example flow size and duration [12]. Using flow features for classification, there is no limitation with respect to packet encryption and privacy concerns, but it is known that many of those techniques are less accurate than payload-based identification techniques [11].

For ML-based classification, there are two types of learning methods, supervised learning and unsupervised learning [13]. Supervised learning creates knowledge structures that support the task of classifying new instances into pre-defined classes, while unsupervised learning (clustering) discovers natural clusters in the data using internalized heuristics [14].

ML-based classification techniques also used for anomaly-based intrusion detection. The authors in [15] proposed a hybrid IDS using Snort [16] with a supervised algorithm to detect attacks. They reported evaluation results with multiple metrics, including accuracy, detection rate, time to build model, and false alarm rate, for different hybrid models. According to their results, Snort with Naïve Bayes algorithm yields promising performance.

A hybrid technique using unsupervised and supervised learning algorithm has also been studied in [17]. The authors grouped similar data instance based on their behavior by using K-Means clustering. Then, the resulting clusters have been classified into attack classes by using a Naïve Bayes classifier as final classification task. If there is any misclassified data during the earlier stage may be correctly classified in the subsequent classification stage.

There was another study evaluated the performance of three well known supervised algorithms: ID3, J48, and Naïve Bayes [18]. The result demonstrates that while Naïve Bayes is one of the most effective inductive learning algorithms, decision trees (J48) are more reliable for the detection of new attacks.

While the above hybrid techniques combined two different approaches, we make a different approach and combine multiple supervised learning algorithms in parallel. This work also considers early detection by utilizing partial flow information. All the above intrusion detection studies [15], [17], [18] used the dataset collected in 1999 by DARPA/MIT [19], but we use a recent dataset collected in 2011[20] as will be discussed in the next section.

TABLE I
Selected Flow Attributes

#	Attribute	Description
1	number-packet	Number of packets in a flow
2	flow-size	Number of bytes in a flow
3	flow-duration	Amount of time that a flow is active
4	avg-packet-size	Average packet size in a flow
5	stddev-packet-size	Standard deviation for of packet size in a flow
6	max-packet-size	Maximum packet size in a flow
7	min- packet-size	Minimum packet size in a flow
8	avg-inter-arrive-time	Average inter-arrival time of packets in a flow
9	stddev-inter-arrival-time	Standard of inter-arrival time of packets in a flow
10	max-packet-inter-arrival-time	Maximum inter-arrival time of packets in a flow
11	min-packet-inter-arrival-time	Minimum inter-arrival time of packets in a flown
12	Packet-size (without handshaking packets)	Size of all packets in a flow without handshaking packets
13	packet-variation	the difference between selected packets in a flow
14	packet-size-mean	Mean of packet size without handshaking packets
15	packet-size-median	Median of packet size
16	application	

III. The Dataset and Flow Features

In this section, we describe the IDS dataset and flow features used in this paper. The dataset is from MACCDC [20], which were collected in 2011 and stored to pcap files [21]. We use all the pcap files that have been collected in 2011. The dataset consists of normal and malicious network flows. The training dataset had 6934 flows. There were 4,481 normal flows and 2,453 malicious flows.

A. Feature Description

From the MACCDC dataset, we considered the following per-flow attributes for classification, also summarized in Table I:

- *Flow Size and Duration:* flow size is the total bytes transferred in a flow. Flow duration is the amount of time between the start and the end of a flow. For TCP, a flow contains a train of packets that begins with a SYN packet and ends with a FIN packet.
- *Packet Size and Number of Packets:* A flow consists of multiple packets, the length of which may be different. We referred to packet information in a flow.
- *Packet Inter-Arrival Time:* we also computed statistics related to packet inter-arrival time, including minimum and maximum inter-arrival seconds, and standard deviation of inter-arrival seconds.

B. Feature Selection

Feature selection is a process that finds the least subset of features, with which we could expect little impact on classification accuracy [22]. As a result, feature selection gives us a positive impact in reducing the time taken to build a ML model, and thus, it is widely used in the ML-based applications. There are several general methods for feature selection: filter method, wrapper method, and embedded method. Among them, we consider the wrapper method since its performance is often acceptable [23]. In the wrapper method, we tested the following three selection techniques: Sequential Forward Selection (SFS) [24], Sequential Forward Floating Selection (SFFS) [24], and PSO Search [25], which are frequently used.

Table II shows the results of feature selection based on the above three wrapper methods. From the table, we can see that some attributes are frequently chosen by multiple methods. In order to optimize the results by the three selection algorithms, we select the *common attributes* that were chosen by every selection method (the last row in Table II). By using only the common attributes, it would be possible to minimize the time and space complexity to run ML algorithms. We observed that using the common set the accuracy is comparable to the one using the entire set of attributes. We will discuss this again in the evaluation section (Section V-B). In the next section, we present our method for intrusion detection that incorporates multiple supervised learning algorithms. The proposed technique uses the common attribute set for its classification.

TABLE II
SELECTED ATTRIBUTES BASED ON FEATURE SELECTION METHODS

Selection method	Attribute	# Attributes
SFS	1, 2, 3, 5, 6, 7, 11, 12, 13	*9*
SFFS	1, 3,4, 6, 7, 8, 10, 11	*8*
PSO Search	1, 2, 4, 6, 7, 8, 10, 11	*7*
Common attributes	1, 3, 6, 7, 11	*5*

IV. INCORPORATING MULTIPLE ML METHODS FOR INTRUSION DETECTION

The simplest form of classification would be to use a single classification method with the common flow attributes. As we will discuss shortly, that would also work well, yielding up to 97.3% accuracy (see Table V for details). However, we observed that it would be possible to improve detection accuracy by incorporating multiple ML algorithms with a simple voting-based selection that chooses a majority. Based on this observation, we develop a detection technique with multiple classification algorithms. In this paper, we consider *three* supervised learning algorithms: AdaBoost [26], J48 Decision Tree [27], and BayesNet [28]. The reason why we selected these three is that they showed stable accuracy in most cases and require relatively low time and space complexity for running. Unlike this, Support Vector Machine requires relatively a great deal of time to obtain a result due to the requirement on a large number of labeled training samples although accurate.

Intrinsically, the classification-based intrusion detection is a *binary* classification: “yes” or “no” to indicate whether the flow in question is malicious or not. For this, the classifiers should be trained with pre-labeled data (labeled as either normal or malicious) for future classification. And based on the cumulated knowledge by training, the classifiers could make a decision against a given input.

Fig. 1 illustrates the procedure of intrusion detection with three ML classification algorithms. For a new input flow, the three ML algorithms run in parallel, each of which produces a binary result. Then, the next step is to combine the individual results to make a final decision: *malicious or not*? As mentioned, we use a simple quorum-based technique. For example, if more than two classifiers say “malicious”, then the flow is classified into illegitimate traffic and a relevant alarm could be raised.

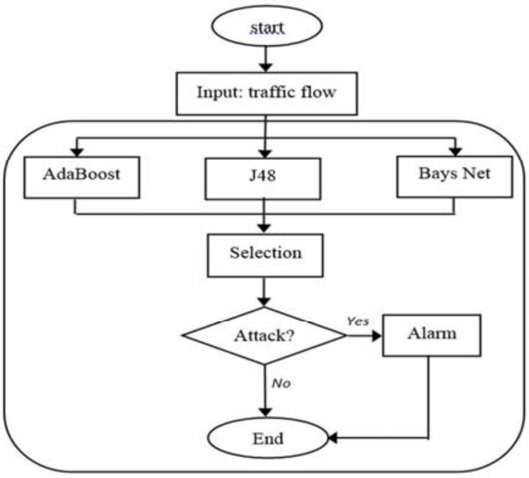
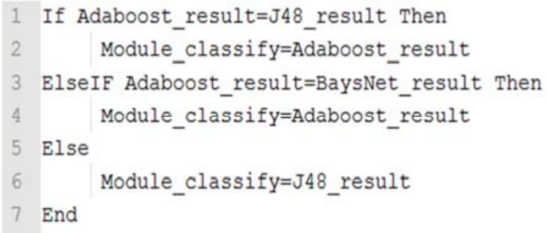


Fig. 1 The procedure of intrusion detection with three ML algorithms



```
If Adaboost_result=J48_result Then
    Module_classify=Adaboost_result
ElseIF Adaboost_result=BaysNet_result Then
    Module_classify=Adaboost_result
Else
    Module_classify=J48_result
End
```

Fig. 2 Combining individual classifier results for making a final decision

Fig. 2 shows pseudo-code for making a final decision from the results by individual classifiers. As shown in the code, if any two classifiers agreed each other against a given flow, we accept the result as the final decision. In this paper, we considered three algorithms, but it could be simply extended with a greater number of ML algorithms since the basic layout of decision is straightforward. In the next section, we provide the experimental results for evaluating our proposed intrusion detection technique.

V. EVALUATION

To evaluate our proposed technique, we used WEKA version 3.7.9 [29] with ten-fold cross-validation. Table III illustrates a matrix with four classes to determine the validity of detection: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). As in the matrix, we call TP if the input flow is actually malicious and the classification result is also malicious. Similarly, it should be FP if the classifier calls malicious although the flow is actually normal.

TABLE III
DETECTION MATRIX

Actual	Predicted Normal	Predicted Malicious
Normal	TN	FP
Malicious	FN	TP

Based on these classes, we use the following metrics to measure the performance of intrusion detection.

- *Accuracy*: the proportion of correct classification classes (i.e., TP and TN) over the total number of classification attempts [30]:

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \times 100\ \%$$

- *Detection Rate*: the percentage of detection out of all scanned inputs:

$$Detection\ Rate = \frac{TP}{TP+FN} \times 100\ \%$$

- *False Alarm*: the proportion of normal traffic flows that are falsely labeled as malicious [30]:

$$False\ alarm\ rate = \frac{FP}{FP+TN} \times 100\ \%$$

As mentioned earlier, it would be beneficial to identify malicious flows as early as possible. We first report our evaluation results for early detection that utilizes partial flow information. We then show the performance with the common attribute set discussed in Section III, by comparing its accuracy with the one with the entire attributes. We finally present the performance of intrusion detection with multiple classifiers.

A. Early Identification with Partial Information

It cannot be stressed enough the importance of early identification of malicious traffic without waiting until the flow is terminated, for immediate report and response. Furthermore, it could reduce the overhead of classification with a smaller set of information processed. However, there could be a trade-off between the classification performance and the amount of information referred to. For example, it would be the best if we could determine whether the flow is malicious or not when its first packet comes in. In that case, however, there must be a lot of false alarms or false negatives.

TABLE IV
ACCURACY RESULTS USING PARTIAL INFORMATION

ML Algorithm	Fist 5 Packets	First 10 Packets	First 15 Packets	All packets
AdaBoost	94%	94.3%	94.6%	95%
J48	96%	96.4%	96.6%	97%
BayesNet	94%	94.5%	94.7%	95%

To see the impact of the number of packets in a flow to accuracy, we conducted a set of experiments that reference a different number of packets for classification: (1) first five packets, (2) first ten packets, (3) first 15 packets, and (4) all packets in a flow. As can be seen from Table IV, there is no significant performance change over the diverse settings. Even only with the first five packets, we can see the maximum difference is equal to 1% for any ML algorithm. If we consider the first 15 packets for classification, the gaps decrease to 0.4% at max.

B. Feature Selection Results

We next present the classification performance when using the common five attributes, the intersection of attribute sets produced by three feature selection algorithms (SFS, SFFS, and PSO Search).

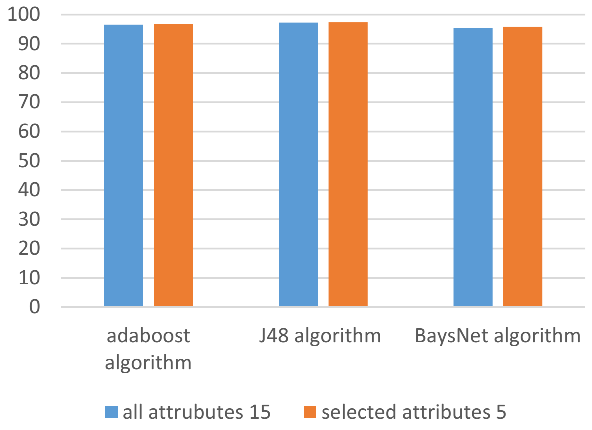


Fig. 3 Comparison Accuracy Results between using all attributes and selected 5 attributes

Fig. 3 compares accuracy between using all attributes and the common attributes. From the figure, we can see no critical performance degradation with the small set of common attributes. And interestingly, we can see even better accuracy results with the small set of selected features.

C. Experimental Results of Intrusion Detection Module

We finally conducted an experiment for intrusion detection. From the above two experimental results, we consider the following in the experiments:

1) Based on the experimental result in Section V *A*, we utilize the first 15 packets instead of waiting until the flow is terminated for early detection of malicious traffic.
2) Based on the result in Section V *B*, we make use of the common attributes in the detection process. Thus, we could reduce the time taken to build ML algorithms.

Table V represents the results of intrusion detection performance for three ML algorithms and our technique. It can be seen clearly that our technique works very well for all the performance metrics including accuracy, detection rate, and false alarm rate. J48 showed the best when using a single ML algorithm. However, our technique outperforms J48 by improving 2% for accuracy with less false alarm rate, archiving greater than 99% accuracy.

TABLE V
COMPARISON ML ALGORITHMS' RESULTS WITH OUR TECHNIQUE FOR INTRUSION DETECTION

Algorithm	TN	FP	FN	TP	Accuracy Result	Detection Rate	False Alarm
AdaBoost	4379	102	121	2332	96.7%	95.8%	2.2%
J48	4448	33	150	2303	97.3%	98.5%	0.7%
BayesNet	4343	138	147	2306	95.8%	94.3%	3.0%
Our technique	4461	22	33	2420	99.2%	99.1%	0.4%

VI. CONCLUSIONS

In this paper, we proposed an effective method that incorporates multiple supervised learning algorithms for intrusion detection. To reduce time complexity and memory requirement, we considered feature selection that selects a subset of attributes for classification, and showed using only five attributes accuracy is comparable to one using the entire attributes or even better. In addition, we explored the number of packets in a flow for early detection, and observed that using the first 15 packets in the flow is a good choice without any significant performance loss. We also developed the procedure for incorporating multiple ML algorithms for detecting malicious traffic, and showed the proposed techniques outperforms non-incorporating techniques that utilizes a single ML algorithm. Specifically, our technique showed over 99% accuracy and detection rate, with partial flow information and attributes. We plan to extend our technique with a greater set of ML algorithms to see the benefits and trade-offs.

REFERENCES

[1] L. Bernaille, R. Teixeira and K. Salamatian, "Early Application Identification," in ACM CoNEXT Conference (CoNEXT '06), 2006.

[2] L. Grimaudo, M. Mellia and E. Baralis, "Hierarchical Learning for Fine Grained Internet Traffic Classification," IWCMC, 2012.

[3] T. E. Najjary, G. U. Keller and M. Pietrzyk, "Application-Based Feature Selection for Internet Traffic Classification," in 22nd International Teletraffic Congress (ITC 2010), 2010.

[4] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos and A. Nucci, "Subflow: Towards Practical Flow-Level Traffic Classification," in INFOCOM, 2012.

[5] V. Paxson, "Bro: A System for Detection Network Intruders in Real-Time," *Computer Network,* no. 31(23-24), pp. 2435-2463, 1999.

[6] V. Kumar, and O. Sangwan, "Signature Based Intrusion Detection System Using Snort," *International Journal of Computer Application & Information Technology*, 2012.

[7] G. Pannell, and H. Ashman, "Anomaly Detection over User Profiles for Intrusion Detection," *Information Security Management Conference*, 2010

[8] C. Pfleeger and S. Pfleeger, *Security in Computing*, 4th ed. Massachusetts U.S.A, 2011, pp 485-486.

[9] J. Eman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/Realtime Traffic Classification Using Semi-Supervised Learning," *Performance Evaluation.,* pp 1194-1213, 2007

[10] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport Layer Identification of P2P Traffic," *the 4th ACM SIGCOMM Conference on Internet Measurement*, pp 121-134, 2004.

[11] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci, "Subflow: Towards Practical Flow-Level Traffic Classification," *Proc IEEE INFOCOM Proceedings - IEEE INFOCOM,* pp 2541-2545, 2012

[12] T. E. Najjary, G. U. Keller and M. Pietrzyk, "Application-Based Feature Selection for Internet Traffic Classification," in 22nd International Teletraffic Congress (ITC 2010), 2010.

[13] T. Nguyen, and G. Armitage, "A Survey of Techniques for Internet Traffic Classification Using Machine Learning," *Communications surveys Tutorials IEEE*, no (10), pp 55-76, 2008.

[14] Y. Reich, J. Fenves, "The Formation and Use of Abstract Concepts in design," *Concepts Formation: Knowledge and Experience in Unsupervised Learning,* 1991.

[15] S. Hussein, F. Ali, and Z. Kasiran, "Evaluation Effectiveness of Hybrid ID Susing Snort with Naïve Bayes to Detect Attacks," *Second International Conference on Digital Information and Communication Technology and its Application*, pp 256-260, 2012.

[16] Snort http://www.snort.org/

[17] Z. Muda, W. Yassin, M.N Sulaiman, and N.I Udzir, "Intrusion Detection Based On K-Means Clustering and Naïve Bayes Classification,"*7th International Conference on (IAS),* pp 192-197, 2011.

[18] M. Panda, M.R. Patra, "A Comparative Study of Data Mining Algorithms for Network Intrusion Detection,"*1st International Conference ICETET,* pp 504-507, 2008.

[19] DARPA/MIThttp://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/

[20] The national cyberWatch Mid-Atlantic CCDC (MACCDC). http://www.netresec.com/?page=MACCDC

[21] Libpcap file format. http://wiki.wireshark.org/Development/LibpcapFileFormat.

[22] M. Dash, and H. Liu, "Feature Selection for Classification," *Intelligent Data Analysis,* pp 131-156, 1997.

[23] S. Gadat, and L. Younes, "A Stochastic Algorithm for Feature Selection in Pattern Recognition," *Machine Learning Research,* pp 509-547, 2007.

[24] G. Ricardo, "CSCE Pattern Analysis," *CSE@TAMU*, 2010.

[25] B. Qinghai, "Analysis of Particle Swarm Optimization Algorithm," *CCSE*, 2010.

[26] R. Schapire, "The Boosting Approach to Machine Learning," *MSRI workshop on Nonlinear Estimation and classification*, 2002.

[27] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[28] P. Cheeseman, and J. Stutz, *Advances in Knowledge Discovery and Data Mining.* Chapter Bayesian Classification: Theory and Result, American Association for Artificial Intelligence, Mentlo Park, CA, USA, 1996, pp 153-180.

[29] WEKA. http://www.cs.waikato.ac.nz/ml/weka/.

[30] S. Wu, and E. Yen, "Data Mining-Based Intrusion Detectors," Expert System with Applications, pp 5605-5612, 2009.

Umar Albalawi is from Saudi Arabia. He got his Bachelor's degree with first class honor in Computer Science from University of Tabuk, Tabuk, Saudi Arabia in 2006. He completed his Master's degree in Computer Science from Texas A&M University-Commerce in 2013. He worked as a Teacher Assistant in the Computer Science and Information Technology Department at University of Tabuk, Tabuk, Saudi Arabia from November 2008 – May 2010. Also, he worked as a Research Assistant in the Computer Science Department at Texas A&M University Commerce From January 2013 – December 2013. He plans to complete his Ph.D. study in Computer and Information Security Filed.