Majority Vote-Based Ensemble Approach for Distributed Denial of Service Attack Detection in Cloud Computing

Ahmed Abdullah Alqarni

Department of Computer Sciences and Information Technology, Al Baha University, Al Baha, Saudi Arabia
E-mail: aaalqarni@bu.edu.sa

Received 03 October 2021; Accepted 23 October 2021;
Publication 16 March 2022

Abstract

Cloud computing is considered as technical advancement in information technology. Many organizations have been motivated by this advancement to outsource their data and computational needs. Such platforms are required to fulfil basic security principles such as confidentiality, availability, and integrity. Cloud computing offers scalable and virtualized services with a high flexibility level and decreased maintenance costs to end-users. The infrastructure and protocols that are behind cloud computing may contain bugs and vulnerabilities. These vulnerabilities are being exploited by attackers, leading to attacks. Among the most reported attacks in cloud computing are distributed denial-of-service (DDOS) attacks. DDOS attacks are conducted by sending many data packets to the targeted infrastructure. This leads to most network bandwidth and server time being consumed, thus causing a denial of the service problem. Several methods have been proposed and experimented with for early DDOS attack detection. Employing a single machine learning classification model may give an adequate level of attack detection accuracy but needs an enhancement. In this study, we propose an approach based on an ensemble of machine learning classifiers. The proposed

Journal of Cyber Security and Mobility, Vol. 11_2, 265–278.
doi: 10.13052/jcsm2245-1439.1126

approach uses a majority vote-based ensemble of classifiers to detect attacks more accurately. A subset of the CICDDOS2019 dataset consisting of 32,000 instances, including 8450 benign and 23,550 DDOS attack instances was used in this study for results and evaluation. The experimental results showed that 98.02% accuracy was achieved with 97.45% sensitivity and 98.65% specificity.

Keywords: Cloud computing, cybersecurity, machine learning, distributed denial-of-service attacks.

1 Introduction

Cloud computing is defined as an internet-based service, enabling the sharing of resources such as storage, network bandwidth, and processing capabilities [1]. It allows the organization and individuals to use resources following the pay-as-you-go approach [2]. Cloud computing provides scalable and reliable services and can be made available over the private, public, or hybrid cloud. To use cloud services, a user must agree and comply with the service level agreement (SLA) of the cloud service provider [3]. The SLA document includes full information regarding the services provided in addition to the security measures. Cloud users are very much apprehensive and concerned about the security and privacy of their data stored in the cloud [4]. Although cloud servers are secured against attacks, there can be situations where an attack may be triggered silently. The dynamic design of cloud platforms breaks the conventional security paradigm used by on-site software programs. Among the reported attacks in cloud computing, distributed denial-of-service (DDOS) attacks are the most common, targeting a cloud infrastructure. DDOS attacks are carried out by exploiting and compromising hundreds of hosts, called zombies, to execute an attack against the target machine. They disrupt the regular traffic on the network through a sudden exponential increase in traffic, clogging network bandwidth and finally preventing the regular traffic from reaching its destination. DDOS attacks have started to grow in scale and complexity, and extortion has been recognized as one of the key factors behind these attacks. DDOS is considered a form of a malicious attack on cloud servers that causes severe problems.

Existing countermeasures for defending DDOS attacks need to classify a data packet into legitimate or malicious [5–7]. Broadly, these methods can be categorized as either signature or anomaly-based. The signature-based attack detection technique involves the use of previously created attack signatures stored in a database. These signatures are matched with captured instances, and if a match is found, then the code is treated as malicious, otherwise legitimate [8]. The downside of using this signature-based detection method is that it cannot find the new malware variant until its signature is not updated in the database. Cybercriminals can use the time since the launch of a new attack and update its definitions into the database to evade detection [9]. Another method used for attack detection is anomaly-based detection. In the anomaly-based attack detection technique, the unusual trends and behavior of a network are determined over a period on the basis of some predefined rules but are more dynamic in nature. If the predefined rules are violated, an alert about the attack is triggered by the system. The downside of this anomaly-based detection method is that it is built manually by professionals but sets up some thresholds. The anomaly-based detection approach is complex, requires a huge amount of time in development, and requires frequent human intervention.

Machine learning has been introduced to overcome the challenges in signature and anomaly-based detection approaches. Although several studies have been conducted using machine learning, applying new machine learning models would continue to be investigated to achieve a higher accuracy level. Machine learning has an intrinsic competency to detect new malware variants on the basis of previous learning, which swiftly helps detect malicious code patterns. This study proposes the ensemble approach to detect DDOS attacks in cloud computing with a high accuracy rate, a low false-positive rate, and negligible performance overheads.

The rest of the paper is organized as follows: Section 2 discusses the studies related to our work. In Section 3, the architecture of the proposed ensemble approach for DDOS attack detection is detailed. In Section 4, the experimental and implementation details and the results obtained are presented. Section 5 concludes the work.

2 Related Work

In the literature, research regarding intrusion detection in computer networks is extensively debated. Multiple approaches have been suggested for DDOS attack detection in cloud computing. A study by [10] proposed a DDOS attack detection approach in cloud computing using several machine learning classifiers. In this study, the authors experimented on their cloud platform. The results show that support vector machines (SVMs) performed better than naive Bayes and random forests. This study's limitation is that it works on a specific type of attack and has several performance overheads in DDOS attack detection. A study by [11] proposed a DDOS attack detection technique using local outlier factor algorithms. These algorithms work by calculating the local variance of a given data point from its neighbors. They attempt to locate anomalous data points. The approach detected the user flood attack with 0.97% accuracy, whereas the Slowloris attack was detected with 0.68% accuracy. A study presented by [12] implemented the K-nearest neighbor (K-NN) algorithm to detect DDOS attacks according to the classification of attack traffic. The downside of this approach is that it has a high false-positive rate and works in offline mode. A study by [13] proposed an approach for detecting DDOS attacks on the basis of the C4.5 algorithm that is used to generate decision trees. In this study, the authors did not mention the features that were used for the classification. A study by [14] proposed the DDOS method using decision trees along with Grey relational analysis. In this study, 15 different features were used to evaluate the incoming and outgoing packets and transmission control protocol synchronization (TCP SYN and acknowledgment (ACK) flag rates to illustrate traffic flow patterns. The selected features and decision trees were employed to detect irregular anomalies in the traffic flow. So far, approaches based on machine learning for DDOS attack detection have proved beneficial in protecting the cloud [15]. In this study, we propose to implement the ensemble approach for DDOS attack detection.

3 Architecture of the Proposed Ensemble Approach for DDOS Attack Detection

The primary objective of this study was to present an approach for DDOS attack detection in cloud computing. This objective was achieved using an ensemble of machine learning classifiers as a methodology for attack detection. The motivation for using the ensemble is to achieve a high rate of detection accuracy with a very low false-positive rate. Machine learning ensemble methods have the property of combining the predictive results of base classifiers and generating one ideal and final predictive model. The ensemble can be of either the same type of classifiers or different types of models. Ensemble models typically yield more precise predictions than a single model would produce. The ensemble of classifiers minimizes the performance overheads and efficient usage of resources. In this study, the ensemble was created using a diverse classifier, namely naive Bayes, decision

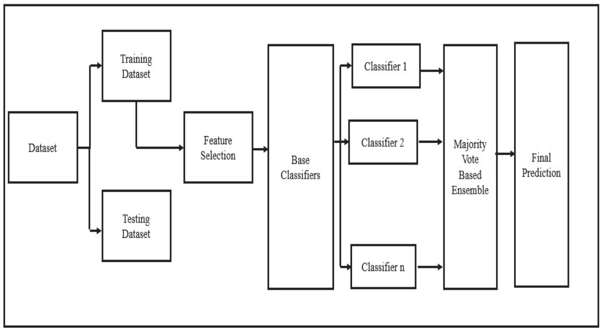

Figure 1 Architecture of the ensemble approach for DDOS attack detection.

tree with the Gini Index, and support vector machine. The selection of these classifiers is based on different machine learning algorithms for DDOS attack detection given in the literature. The output of the classifiers is combined through majority voting. The proposed approach uses a dataset, CICD-DOS2019, to discover and investigate the hidden information associated with DDOS attacks to accurately distinguish between malicious and benign codes. The proposed approach can help security solution providers make appropriate decisions that are not possible through traditional DDOS attack detection approaches. Figure 1 depicts the architecture of the proposed approach. The approach is divided into four sections: data collection, feature selection, base classifiers, and ensemble.

3.1 Data Collection

This study obtained a dataset called CICDDOS2019 from the Canadian Institute of Cybersecurity at the University of New Brunswick [16]. The institute provides a complete DDOS attack dataset for research purposes. The dataset consists of approximately 1 million benign and 30 million malicious instances of traffic flow. The malicious instances are classified into 13 categories. It has 80 features that are related to network traffic flow. As the CICDDOS2019 dataset is huge, it was necessary to limit the size of the dataset used in this study. Division of the dataset may lead to the overfitting problem. We used two methods known as under- and oversampling to have a balanced dataset to handle the overfitting problem. The extracted dataset used in this study consists of 32,000 instances, with 8450 benign and 23,550 DDOS attack instances.

3.2 Feature Selection

Feature selection is a method of reducing the input variables to generate a predictive model [17]. The reason for feature selection is that not all features contribute to the accuracy of the model. The original dataset consists of 80 features; to reduce the number of features, use only those that are very much relevant. In this study, we used the chi-squared feature selection method and selected only the top 15 features. Since the chi-squared method is statistical, it works to determine the observed "O" and expected "E" distance among the variables [18]. The distance determines the correlation between the variables; if the distance is stronger, the correlation is high. The formula for calculating the chi-square for each feature is given in Equation (1). Features with high dependence on response are selected, and the list of features is provided in Table 1.

$$\mathcal{X}_c^2 \sum \frac{(O_{i-}E_i)^2}{E_i} \tag{1}$$

where c = degree of freedom, O = observed value(s), E = expected value(s).

3.3 Base Classifier

To evaluate the proposed DDOS attack detection approach, we used four different classifiers belonging to different categories, and the ensemble is based on these classifiers using majority voting. The classifiers used in this study are provided below.

Table 1 Features used in this study

	Features		Features
1	Average packet size	9	Variance of packet length
2	Median packet length	10	Median packet time
3	Mode packet length	11	Mode packet time
4	Protocol (TCP or UDP)	12	Variance of packet time
5	Flow duration (duration of the flow in milliseconds)	13	Standard deviation of packet time
6	Min packet length (minimum length of a packet)	14	Coefficient of variation of packet time
7	Max packet length (maximum length of a packet)	15	Skew from median packet time
8	Standard deviation of packet length		

3.3.1 Naïve Bayes

Naïve Bayes classifiers form a probabilistic model that is based on Bayes' theorem [19]. These classifiers have the property of simple learning by considering that features are independent given the class. The naive Bayes classifier assigns a most likely class to an instance on the basis of its feature set. Naïve Bayes classifiers have emerged as an essential tool with many practical uses, including text classification, medical record interpretation, and malware detection. The naive Bayes classifier is represented as given in Equation (2), [8].

$$p(cj \mid d)) = \frac{p(d \mid cj)p(cj)}{p(d)} \tag{2}$$

Where,

$p(cj \mid d) =$ probability of instance d being in class cj
$p(d \mid cj) =$ probability of generating instance d given class cj
$p(cj) =$ probability of the occurrence of class cj
$p(d) =$ probability of instance d occurring

3.3.2 Decision trees

The decision tree, a machine learning classifier, is a predictive model that is based on decision trees. Decision trees use a tree-like model for decision-making and deciding the possible outcome of an event. In a tree structure, the leaf nodes are assigned to class labels. The root and internal nodes include feature test conditions to classify an instance that has distinct characteristics [20]. Decision tree classifiers are highly considered to have intelligibility and simplicity. The downside of using a decision tree is that it may encounter an overfitting problem when the tree is fully grown, leading to the loss of certain generalization capabilities. The overfitting problem mostly arises because of the occurrence of noise and improper representations of instances. This problem is solved through repruning and post pruning methods.

3.3.3 Support vector machines

SVMs have turned out to be one of the standard tools for machine learning and data mining. They belong to the supervised learning category [21]. SVMs are widely used for the classification and regression analysis of data. Vapnik developed SVMs at AT&T Bell Laboratories [22]. They are considered a highly robust prediction model that is centered on statistical learning frameworks. Using the SVM classifier, the classification is conducted by developing an N-dimensional hyperplane that optimally divides the data into two classes and increases the hyperplane distance. SVMs are also used when data is not linearly separated by employing kernel functions for data separation.

3.3.4 K-Nearest neighbor

K-Nearest Neighbor (K-NN) is one of the simplest machine learning models commonly applied in classification and regression [8]. It is usually used for its straightforward interpretation and has low computation time. The selection of the parameter *k* is extremely vital in this model. Two parameters, such as the training and validation error rates, are applied to distinct *k* values. On the basis of the similarity, K-NN creates new data points from stored points. It does not make any assumptions on the underlying data and is considered a nonparametric algorithm. The K-NN algorithm only holds a dataset at the testing level, and when it receives new data, it classifies the data into a group that is somewhat close to the new data.

3.4 Ensemble by Majority Vote

Majority voting is regarded as one of the simpler and most efficient forms of combining the predictions produced by different classifiers. In majority voting, each class's votes are counted over the input classifiers, and the majority class is chosen [23]. The principle of choosing and selecting classifiers is to create an ensemble rather than use all classifiers experimented with in various ways. There are three types of majority voting from which the ensemble can, first, choose the class "unanimous vote" in which all classifiers agree; second, the simple majority in which the same prediction is made the latest by 50% of classifiers; and third, plurality voting. The instance gets the highest number of votes whether or not the total sum of these exceeds 50%. In this study, we used majority voting.

As shown in Figure 2, suppose we have a training set, and a set of classifiers as $h_1, h_2, \ldots, h_n$, and each classifier was trained on a training set. So, after training, the classifier will produce predictions. The classifier h_1 will produce the prediction y_1; classifier h_2, prediction y_2; and classifier h_n, prediction y_n. For every new data point, we have *n* predictions. Then, we can have voting to arrive at the final prediction. Voting reduces *n* class label predictions for a single data point into a single class. Therefore, we used majority voting to decide the final vote. Mode operation is used to obtain the

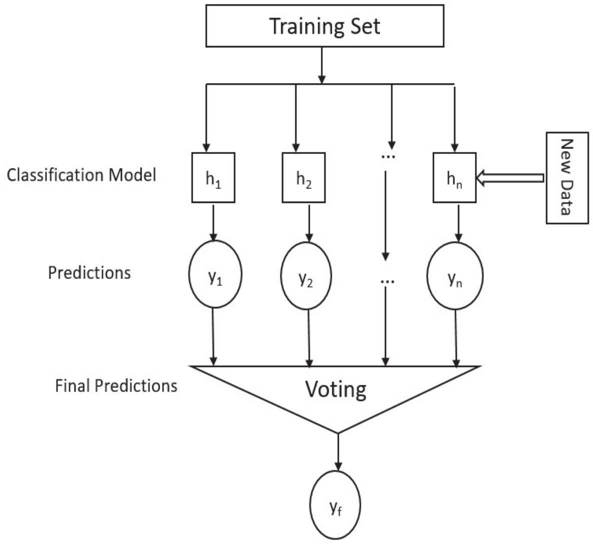

Figure 2 Majority vote-based ensemble approach.

final vote, which is expressed as in Equation (3).

$$y_f = \text{mode}\{h_1(x), h_2(x), \ldots, h_n(x)\} \tag{3}$$

where $h_i(x) = y_i(x)$.

Using majority voting is like asking a committee of experts to vote on a specific resolution. If the majority of members give their vote, then the resolution is accepted. In this way, the chances of making prediction errors are negligible.

4 Experimental Results

The experiments were conducted on a stand-alone computer with i5, 3.5Ghz processing capabilities, and 8GB RAM. The DDOS attack dataset was evaluated using Weka (developed by the University of Waikato, New Zealand). Weka is an open-source tool built using Java and is used widely in machine learning. The advantages of Weka include its flexibility, processing speed, and user-friendly interface. To divide the dataset into training and testing data, we used a cross-validation scheme. Cross-validation performs the resampling of data given to a machine learning model for producing predictions on unseen data. In this study, we used *k*-fold cross-validation, where the value

Table 2 Results were obtained using different classifiers and the ensemble approach on the dataset

Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)	Execution Time Per Instance (Milliseconds)
Naïve Bayes	96.24	92.33	92.62	0.7
Decision tree	95.33	92.69	95.87	0.8
SVM	95.85	93.02	97.48	0.4
K-NN	96.01	92.15	95.88	0.6
Ensemble	98.02	97.45	98.65	3.2

of $k = 5$ was used. Through this, we achieved multiple test and train splits to generate unbiased results. The experimental results of the proposed approach were evaluated using four measures: accuracy, sensitivity, specificity, and execution time.

Accuracy is considered one of the basic performance evaluation metrics through which the classification accuracy of the DDOS attack detection approach is determined. High detection accuracy means that the system is highly able to detect attacks. If the achieved detection accuracy is low, then the method has not achieved the objective. Sensitivity determines the number of instances classified correctly as benign in the dataset. Specificity implies the number of instances classified correctly as attacks. Execution time shows the amount of time taken by each classification model in producing the prediction results. Mathematical accuracy, sensitivity, and specificity are given in Equations (4), (5), and (6), respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

In this study, the first four classifiers—such as naive Bayes, decision tree, SVM, and K-NN—were used, and the ensemble of the classifiers was generated using majority voting. The results obtained are given in Table 2.

The experimental results generated using the CICDDOS2019 dataset are presented in Table 1. In the first step, the dataset was checked for consistency of the data. Since the dataset was huge, a subset of the dataset was extracted to keep the imbalance problem to avoid over-and underfitting data. The original

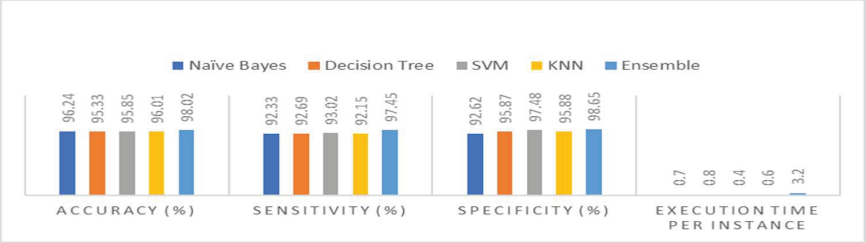

Figure 3 The performance comparison.

CICDDOS2019 dataset has 80 features, but we selected only 15 features through the chi-squared method. Cross-validation of 80:20 was used to split the data. The classification was done using Weka, selecting all 15 features. First, we evaluated individual classifiers—such as naive Bayes, decision tree, SVM, and K-NN—by evaluating the ensemble of these classifiers using majority voting. The comparison of the experimental results, as illustrated in Figure 3, showed that our ensemble approach performed much better than did individual classifiers with 98.02% accuracy, 97.45% sensitivity, and 98.65% specificity. However, there was a slight increase in execution time, which was 3.2 milliseconds, but this is considered very normal in ensembles.

5 Conclusion

In this study, we proposed and implemented the ensemble approach for DDOS attack detection. The ensemble works on majority voting, and naive Bayes, decision trees, SVMs, and K-NN were used as base classifiers. DDOS attack detection is a broad topic of research, and in this study, our focus was DDOS attack detection in cloud computing. The experimental results showed that our proposed ensemble approach performed better than did individual classifiers with a reduced dataset and limited the feature size. Finally, our future work will focus on exploring the implementation of deep learning in DDOS attack detection.

References

[1] Hu P, Dhelim S, Ning H, Qiu T. Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications. (2017), 15;98:27–42.

[2] Chaudhary D, Bhushan K, Gupta BB. Survey on DDoS attacks and defense mechanisms in cloud and fog computing. International Journal of E-Services and Mobile Applications, (2018) 1;10(3):61–83.

[3] Zhou H, Ouyang X, Ren Z, Su J, de Laat C, Zhao Z. A blockchain based witness model for trustworthy cloud service level agreement enforcement. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications (2019) Apr 29 (pp. 1567–1575).

[4] Jayaraman I, Panneerselvam AS. A novel privacy preserving digital forensic readiness provable data possession technique for health care data in cloud. Journal of Ambient Intelligence and Humanized Computing. (2021), 12(5):4911–24.

[5] Amjad A, Alyas T, Farooq U, Tariq MA. Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm. EAI Endorsed Transactions on Scalable Information Systems. (2019), 6(26).

[6] Zekri M, El Kafhali S, Aboutabit N, Saadi Y. DDoS attack detection using machine learning techniques in cloud computing environments. In IEEE 3rd international conference of cloud computing technologies and applications (CloudTech) (2017) Oc (pp. 1–7).

[7] Vimala S, Dhas J. SDN based DDoS attack detection system by exploiting ensemble classification for cloud computing. International Journal of Intelligent Engineering and Systems. (2018) 11:282–91.

[8] Khan N, Abdullah J, Khan AS. Defending malicious script attacks using machine learning classifiers. Wireless Communications and Mobile Computing (2017) 5360472.

[9] Khan N, Abdullah J, Khan AS. A Dynamic Method of Detecting Malicious Scripts Using Classifiers. Advanced Science Letters. (2017), 23(6):5352–5.

[10] Wani AR, Rana QP, Saxena U, Pandey N. Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In IEEE Amity International conference on artificial intelligence (2019), (pp. 870–875).

[11] Madhupriya G, Shalinie SM, Rajeshwari AR. Detecting DDoS attack in cloud computing using local outlier factors. In IEEE 2nd International Conference on Trends in Electronics and Informatics (2018), (pp. 859–863).

[12] Xiao F, Ma JQ, Huang XS, Wang R. DDoS attack detection based on KNN in software defined networks. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition). (2015), 35(1):84–8.

[13] Zekri M, El Kafhali S, Aboutabit N, Saadi Y. DDoS attack detection using machine learning techniques in cloud computing environments. In IEEE 3rd international conference of cloud computing technologies and applications (CloudTech) (2017) Oct 24 (pp. 1–7).

[14] Wu YC, Tseng HR, Yang W, Jan RH. DDoS detection and traceback with decision tree and grey relational analysis. International Journal of Ad Hoc and Ubiquitous Computing. (2011), 7(2):121–36.

[15] Khan N, War TA. A Deep Study on Security Vulnerabilities in Virtualization at Cloud Computing. International Journal of Computer Applications. 975:8887.

[16] Canadian Institute of Cybersecurity, University on New Brunswick https://www.unb.ca/cic/datasets/index.html accessed on 17/01/2020.

[17] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of machine learning research. (2003)1157–82.

[18] Sharpe D. Chi-square test is statistically significant: Now what?. Practical Assessment, Research, and Evaluation. (2015), 20(1).

[19] Berrar D. Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. Elsevier Science Publisher: Amsterdam. The Netherlands. (2018),1:19.

[20] Kang Q, Shi L, Zhou M, Wang X, Wu Q, Wei Z. A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification. IEEE transactions on neural networks and learning systems. (2017) 25;29(9), pp. 4152–65.

[21] Khosravi K, Pham BT, Chapi K, Shirzadi A, Shahabi H, Revhaug I, Prakash I, Bui DT. A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at Haraz watershed, northern Iran. Science of the Total Environment. (2018) 627 pp. 744–55.

[22] Ordóñez C, Lasheras FS, Roca-Pardinas J, de Cos Juez FJ. A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. Journal of Computational and Applied Mathematics. (2019), 346, 184–191.

[23] Bai J, Wang J. Improving malware detection using multi-view ensemble learning. Security and Communication Networks. (2016)9(17) pp. 4227–41.

Biography

Ahmed Abdullah Alqarni received the bachelor's degree in computer science from King Abdulaziz University in 2004, the master's degree in information technology from La Trobe University in 2010, and the philosophy of doctorate degree in computer science from La Trobe University in 2016, respectively. He is currently working as an Assistant Professor at the Department of Information Technology, Faculty of Computer Science and Information Technology, Al Baha University. His research areas include Cyber Security, Machine Learning, and Artificial Intelligence. He has been serving as a reviewer for many highly respected journals.