HYBRIDIZATION OF SUPPORT VECTOR MACHINE WITH CAT SWARM ALGORITHM FOR INTRUSION DETECTION

By

I. O. OYEFOLAHAN * J. N. NDUNAGU ** S. IDRIS *

* School of Information and Communication Technology Federal University of Technology Minna, Nigeria.
** Department of Computer Science, National Open University of Nigeria, Abuja, Nigeria.

Date Received: 17/07/2020 Date Revised: 09/09/2020 Date Accepted: 15/12/2020

ABSTRACT

Intrusion detection system (IDS) like firewall, access control and encryption mechanisms no longer provide the much-needed security for systems and computer networks. Current IDS are developed on anomaly detection which helps in identifying attacks both known and unknown. Unfortunately, these anomaly-based IDS feature high false rate. In a bid to reduce this false alarm rate, this paper proposed an intrusion detection model based on Support Vector Machine (SVM) optimized with Cat swarm optimization (CSO) algorithm. Attribute reduction was carried out based on Information Gain (IG) and classification was performed based on the optimized Support vector. The result obtained shows that our model performs well with the least false alarm rate and good accuracy value compare with other classification algorithms evaluated using the same datasets.

Keywords: Ntrusion Detection, Support Vector Machine, Cat Swarm Optimization, Information Gain, NSL-KDD.

INTRODUCTION

One of the major technological achievement in recent time is the possibility of connecting computer systems for the purpose of sharing resources. Furthermore, the advent of the internet has made it possible for people to communicate from different part of the globe through connected computer networks. However, these interconnection of computer devices came with its own cons. One of the major issues with this technology is in the area of security. computer networks and the internet at large are faced with many security attacks. These attacks aim to compromise the confidentiality, integrity and availability of the network and its resources.

Many protection techniques have been employed to manage the security risks involved with computers and networks. Techniques like encrypting confidential data, access control and software and hardware firewall policies. However, these methods are not enough as each of them possess significant limitations. Therefore, it becomes important to use other additional approaches to defense of which intrusion detection system (IDS) is a viable example (Enache & Patriciu, 2014). IDS is a software application or device that monitors computing platforms connected by networks for abnormal activities and report or prompt for appropriate action (Kumar et al., 2013). Many researches have been carried out by researchers to find an intrusion detection technology with good detection accuracy and less training time. However, many issues still exist with IDS, such as low detection capability, high false positive rate (Kuang et al., 2014).

Many techniques have been proposed to improve the performance of IDS in recent times. Support Vector Machine (SVM) is one of such techniques. SVM is a new machine learning method that has become a well-known research method in the area of intrusion detection. This is because of its good generalization performance, unavailability of local minimal and fast execution time (Enache & Patriciu, 2014). According to Enache and Patricia (2014), SVM performance is determined by the selection of appropriate parameters for it. Failure to ensure that appropriate parameters are selected, the technique will perform poorly. Thus, this study proposed an approach

(system) to intrusion detection by optimizing SVM with Cat Swarm Optimization.

1. Intrusion Detection System

An intrusion detection system (IDS) is a software application that checks network activities for suspicious or malicious processes and generates reports for necessary action(s). A network connectivity increases, intrusion detection has become an important network security measure (Al-Yaseen et al., 2017). Traditional methods like encryption, firewalls or access controls are no longer capable of providing full protection as malicious activities continue to take different dimensions (Hassan et al., 2017). Therefore, "an intrusion detection system has become an important component of security infrastructure and a key part of system defense to detect these attacks before such attacks lead to disaster in the system" (Hassan et al., 2017). Consequently, an IDS attempts to flag malicious activities and usages that appear compromised on the network and which the traditional method is incapable of detecting. A trending approach for IDS development has been based machine learning concepts. Specifically, detection approaches that have been based on hybridized-learning techniques have performed better compared to mono-learning techniques or approaches (Lin et al., 2015).

IDS can be categorized in different ways. The most common categories or types are misuse and anomaly intrusion detection systems (Aljawarneh et al., 2018; Enache & Patriciu, 2014).

1.1 Misuse-based Intrusion Detection System

This is also referred to as signature-based intrusion detection system. In misuse or signature-based intrusion detection, packets or logs are scanned to look for activities which fall into the category of attacks that have been previously determined (Lin et al., 2015). Misuse-based IDS can efficiently detect known attacks. This type of IDS has a low false alarm rate, but it fails to identify new attacks that are not familiar or do not embody any rules in the database (Al-Yaseen et al., 2017; Aljawarneh et al., 2018)

1.2 Anomaly-Based Intrusion Detection System

Anomaly IDS is based on normal behavior parameters and utilizes them to identify activities that are in dis-consonance with normal behavior. It identifies patterns based on the examination of data taken from normal usage (Aljawarneh et al., 2018). Anomaly IDS identifies malicious activities based on deviation from what is known to be normal behavior (Lin et al., 2015).

2. Categories or Classes of Intrusion Attacks

Intrusion attacks can be classified into: Remote to local (R2L), User to Root (U2R), Probing and Dos or DDoS attacks (Dhanabal & Shantharajah, 2015).

2.1 Remote to Local Attacks (R2L)

A Remote to Local attack which is also referred to as Remote to User attack is carried out when an attacker transmits packets to targeted host over a network with the intention of revealing the vulnerabilities of the host which will enable the attacker to exploit the privileges of a local user (Alharbi et al., 2018; Raiyn, 2014).

2.2 User to Root Attack (U2R)

In user to root attack, the attacker ensures that he gain a footing on the remote system which makes it look like a normal user session. By using variety of techniques, the attacker attempts to increase his privileges incrementally until he gains root access on the remote system (Alharbi et al., 2018; Aljawarneh et al., 2018).

2.3 Probing Attack

A probing attack is usually carried out with the intention of letting the target system detects it and from the report generated by the targeted system, the attacker uses that as prelude to detect the system's location, defensive capabilities and vulnerabilities. Ideally, these attacks are aimed at revealing information about the defensive weaknesses in the targeted systems. (Alharbi et al., 2018; Raiyn, 2014).

2.4 Dos/DDoS Attack

Denial of Service (DoS) attacks are intended to render unavailable the resources of a targeted system. Thus, making it unusable to legitimate users. These attacks are achieved by sending traffics with volumes greater than the capability of the targeted system, thus making the system suspend operation to legitimate users or crashing its operations entirely. (Alharbi et al., 2018; Desai et al., 2017).

When many compromised system are involved in DoS attack, it is referred to as distributed denial of service attack (DDoS) (Hoque et al., 2017; Nidhi & Prasad, 2016; Yadav et al., 2016). DoS attacks can occur at different layer of the open system interconnection model like the physical layer, network layer, transport layer and application layer (Tan et al., 2013; Yadav et el., 2016). At the physical layer, the attack can be carried out simply by removing power or network cable connecting a server to the network. Attacks at the network layer can be carried out via network layer protocol like the Internet Control Message Protocol (ICMP) (Singh & De, 2017). At the transport layer, the attacks can be achieved using layer four protocol like the user datagram protocol and the transmission control protocol (TCP) (Singh & De, 2017). DoS attacks can be carried out at the application layer via protocols like Hypertext Transmission Protocol (http), Simple Mail Transmission Protocol (SMTP), Domain Name System (DNS), Voice over Internet Protocol (VoIP).

2.4.1 Types of DoS attack

DoS and DDoS attacks are classified according to the resources they consumed. Some attacks are targeted at consuming the network bandwidth, these attacks are called network depletion attack. In another category of attack, the attacker depletes the system resources like the system memory and CPU, this is called resource depletion attack. Other categories are infrastructure and zero-day attack (Mahjabin et al., 2017)

- Resource Depletion Attack

These are attacks targeted at completely exhausting the resources of the system such as the system memory and central processing unit (CPU). Resource depletion are usually carried out in two ways: protocol exploit attack and malformed packet attack. Protocol exploit involves protocol-based attack that exploit the flaws of various network layer protocols, forcing the attacked system to exhaust all its CPU and memory while carrying out memory intensive operations. Examples of protocols that can be exploited are Transmission Control Protocol (TCP), Hyper Text Markup Language (HTTP) and Session Initialization Protocol (SIP). A malformed packet attack is an attack carried out using a packet that is deformed so as to confuse the victim (Mahjabin et al., 2017).

- Infrastructure Attack

This is an attack carried out on both the bandwidth and other resources (CPU and memory). This attack aims to damage important part of the information super-highway. Example of protocol that this attack usually aim is the Domain Name System (DNS). Infrastructure attack can be carried out using compromised systems to send normal User Datagram Protocol (UDP) request to the DNS server (Mahjabin et al., 2017).

- Zero-Day Attack

This is an attack that is unknown to those who could have patched it. A zero-day attack uses vulnerability found on target systems in day one to attack systems.

- Bandwidth Depletion attack

In this attack, the victim bandwidth is flooded with unwanted traffics, these unwanted traffics prevent legitimate traffic from accessing the targeted system (Devare et al., 2016). Examples of protocols used in carrying out network depletion attack are the user datagram protocol and the internet control message protocol (Hoque et al., 2017; Harshita, 2017). According to the prolexic (the word largest DDoS service) Among the protocols commonly used to launch DDoS attacks, TCP, UDP and ICMP (Saied et al., 2016).

- ICMP Flood Attack

The internet control message protocol (ICMP) is a connectionless oriented protocol used for network connectivity testing, error reporting, and querying server (Desai et al., 2017); Daş et al., 2015). Because of this important function that ICMP is used for, it is now being targeted as an attacking tool by system and network attackers. ICMP is used to compromise system and network through the ICMP flood attack. In ICMP flood attack, the adversary sends ICMP ECHO-REQUEST traffic to a network of compromised systems usually called botnet, the compromised systems send back ECHO REPLY message to the spoofed source address. The ECHO_REPLY message overwhelms the victim system making it to drop packet from legitimate users (Yadav et al., 2016; Mahjabin et al.,

2017; Harshita, 2017).

Available techniques used for preventing or mitigating ICMP flood attack employs general purpose firewall and intrusion detection systems (Desai et al., 2017; Daş et al., 2015). Using general firewall and intrusion detection system have not help much to prevent or mitigate ICMP flood attack especially if the attack is coming from a formerly trusted network that has been compromised.

3. Support Vector Machine

Support vector machine is a machine learning algorithm that has gained importance in the area of pattern classification. SVM primarily aim at finding an optimum hyperplane to separate two classes in a dataset. Several machine learning algorithms exist for dataset classification, SVM standout of these algorithms because of its outstanding generalization capability and reputation in the training data set to achieve high accuracy (Enache & Patriciu, 2014)

Classification problem has several major challenges, one of them is data dispersion, the dataset is not always linearly separable. To overcome this issue of linearly inseparable datasets, the dataset can be mapped onto higher-dimension feature space in order to find the hyperplane that separates the mapped vectors linearly. Thus, x_i will be replaced with (x_i) where K (kernel function) gives the higher dimensional mapping. Usually, kernel functions have several forms of which two are briefly described below: polynomial kernel function and radial-basis kernel function (RBF) (Enache & Patriciu, 2014).

3.1 Polynomial Kernel Function

The Polynomial kernel is a non-stationary kernel that is commonly used with SVM. It is appropriate for problems with normalized training samples (Amami et al., 2015). The polynomial function is represented as follows.

$$K(x_i, x_j) = (\sigma x^T{}_i x_j + r)^d, \sigma > 0 \qquad (1)$$

3.2 Radial-Basis Kernel Function (RBF)

RBF (Gaussian) kernels is a kernel that takes the form of radial basis function. This kernel non-linearly maps samples into a higher dimensional space. So, it is unlike the linear kernel, it can handle the case when the relation between class labels and attributes is nonlinear. (Amami et al., 2015). The RBF kernel is represented as follows.

$$K(x_i, x_j) = \exp(-\sigma \left\| x_i - x_j \right\|^2), \sigma > 0 \qquad (2)$$

Where the parameter σ set the "spread" of the kernel.

4. Cat Swarm Optimization

One of the types of optimization problem is feature selection. It is usually achieved by combining an optimization algorithm with a classification algorithm. Two commonly used optimal algorithms are Particle Swarm Optimization (PSO) and genetic algorithm (GA). Cat swarm optimization (CSO) has recently been proposed and demonstrated to perform better compare to PSO (Lin et al., 2016; Hadi & Sabah, 2015). CSO was built based on t he behavior of cats, which are excellent hunters but which also exhibit high levels of alertness even when at rest. This cat behavior can be described by two modes: "Seeking and Tracing modes" PSO (Lin et al., 2016; Kumar et al., 2016).

4.1 Seeking Mode

This mode describes the situation of the cat while resting. In this mode, the cat thinks and decides about next move (Hadi & Sabah, 2015). The seeking mode are represented in the CSO algorithm by four parameters: Seeking memory pool (SMP), Seeking Range of the selected dimension (SRD), count of dimension to change (CDC) and Self-Position Consideration (SPC) (Bahrami et al., 2018). The process of seeking mode is described below:

"Step1: Make j copies of the present position of catk, where j = SMP. If the value of SPC is true, let *j* = (SMP-1), then retain the present position as one of the candidates.

Step2: For each copy, according to CDC, randomly plus or minus SRD percent of the present values and replace the old ones.

Step3: Calculate the fitness values (FS) of all candidate points.

Step4: If all FS are not exactly equal, calculate the selecting probability of each candidate point by (4), otherwise set all the selecting probability of each candidate point be 1.

Step5: Randomly pick the point to move to from the candidate points, and replace the position of cat_k."

$$Pi + \frac{FSi - FSb}{FS\max - FS\min} where\ 0 < i < J \qquad (3)$$

If the fitness function goal is to find the minimum solution, FSb = FSmax, otherwise FSb = FSmin.

4.2 Tracing Mode

This mode describes the situation of the cat while chasing a target. A cat in a tracing mode moves according to its own velocity for every dimension (Bahrami et al., 2018). The tracing mode process is given below:

"Step1: Update the velocities for every dimension (vk,d) according to (5).

Step2: Check if the velocities are in the range of maximum velocity. In case the new velocity is over-range, set it be equal to the limit.

$$V_{k,d} = V_{k,d} + r_1 x C_1 (X_{best,d} - X_{k,d}) \qquad (4)$$

Step3: Update the position of cat_k according to (5)".

$$X_{k,d} = X_{k,d} + V_{kd} \qquad (5)$$

x_{best} is the position of the cat, with the best fitness value; $X_{k,d}$ is the $cat_{k\ position}$. c_1 and r_1 is a constant and random value respectively and r_1 has a value range of [0,1].

5. Related Work

Kevric et al. (2017) developed a network intrusion detection algorithm. The algorithm combined two tree-based classifier models; the random tree and Naive bayes tree classifiers. The paper aim is to have a hybrid classifier that can classify traffic entering a network into normal or attack with better accuracy compare to the individual classifiers. The study used the NSL-KDD dataset to evaluate the performance of their classifier. Detection accuracy of 89.24% was achieved. The study proposed testing effect of feature reduction both on the training and detection accuracy as future work.

Also, Saied et al. (2016) proposed a framework for detecting and addressing issues of known and unknown distributed denial of service in real time using artificial neural network. The study used ANN to detect attack based on some features that distinguishes DDoS attack from normal attack. The ANN was trained with data collected from a network setting that represented a mirror image of a real-life network environment. In addition to the data collected from the mirror network, the study used old data to evaluate their work. A detection accuracy of 98% was recorded. The future work would be to train their approach using other dataset and compare the outcome with the outcome they got. Also, their work was not simulated in any network environment, one could simulate their approach to verify the detection accuracy of the work and the false alarm rate.

A multilayer perception with genetic algorithm for detecting DDoS at the application layer was proposed by Singh and De (2017). Four features were considered from incoming traffic that exhibit significant changes in their characteristics. The http count is the first parameter and its features like the GET, POST, OPIONS, HEAD, DELETE, PUT, TRACE, and CONNECT were analyzed and normal features where recorded. The count of IP address that enters a network within a space of time is the second parameter. The third parameter is the constant mapping function. The fourth parameter is the fixed frame length. When there is a change in these features, attack will be detected. Experiment result show that the method gives an accuracy of 98.04% detecting at the application layer; but with a high false positive rate of 2.21%. The future work is to work on improving the detection accuracy and lowering the false positive rate.

In another study, Enache & Patriciu (2014) proposed an IDS based on support vector machine combined with information gain. They used Information Gain to select the features of the dataset and the SVM was used for classification. The parameters for Support Vector Machine were selected based swarm intelligence-based algorithm (Particle Swarm Optimization or Artificial Bee Colony). The performance of the algorithm was tested using NSL-KDD dataset. It was found that the SVM optimized with either particle swarm optimization or artificial bee colony performed better compared to the normal SVM. The future work would be to apply swarm intelligence data mining algorithms for the feature selection process hoping to obtain better result. Also, other swarm intelligence algorithms can be explored as well.

Ashfaq et al. (2017) used fuzziness based semi supervised approach to investigated an IDS by supporting unlabeled samples with supervised learning algorithm improve the

performance of the classifier. The results of this technique on NSL-KDD dataset shows better output compared to existing classifiers like naive bayes, support vector machine, random forest. They got an accuracy of 84.12%. The future work proposed is to apply this strategy to improve the effectiveness of IDSs for detecting multiple types of attacks.

6. Proposed Model

Figure 1 depict our model. Acquiring the NSL-KDD dataset, Preprocessing and attribute selection are pre-requisite in any intrusion detection work. Our contributions begin with optimizing the parameters of the SVM using Cat Swarm Optimization Algorithm.

6.1 Data Processing/Feature Selection

The model was evaluated using the NSL KDD data set. The dataset is an improved version of the KDD dataset. It has numerous advantages over the KDD Cup 99. The advantages include: absent of redundant record in the train dataset, no duplicate records in the test dataset. The NSL KDD dataset consist of 41 features. However not all of the features that are relevant. Therefore, the need for feature selection. In other to carry out feature selection, we use information gain (Entropy).

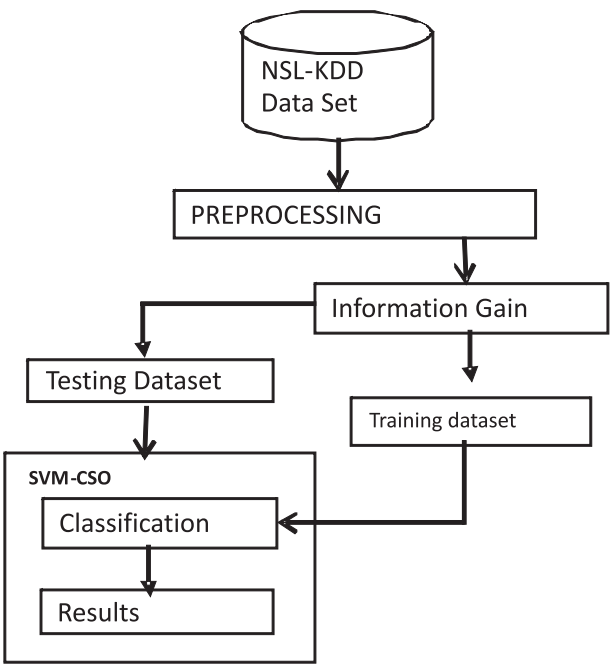

Figure 1. Proposed Model

Entropy is a common criterion used in machine learning to individually rank features with respect to class attributes. IG is measured by a reduction in the uncertainty of ascertaining the class attribute when the feature value is not known. Its idea uses the information theory approach for ranking and selecting top features so as to achieve improved classification with less complexity.

The IG of a given attribute A with respect to the class attribute C, denoted as I(C/A), is the reduction in uncertainty about the value of C when the value of A is known. Let C and A be discreet variables that take the values $C = (c_1, \ldots, c_k)$ and $A = (a_1, \ldots, a_n)$. H(C) is the entropy, that measures the uncertainty in the value of C. H(C/A) is the conditional entropy of C given A, that measures the uncertainty on the value of C after observing values of A. Thus I(C/A) = H(C) - H(C/A) [1].

$$H(C) = -\sum_{J=1}^{k} P(Ci)\log 2(P(Ci))$$
$$H(C/A) = -\sum_{j=1}^{n} P(aj)\sum_{i=1}^{k} P\left(\frac{Ci}{aj}\right)\log 2(P(Ci/aj)) \quad (6)$$

where, P(ci/aj) is the posterior probabilities of C given the values of A. therefore I(C/A) is given as follows.

$$I(C/A) = -\sum_{I=1}^{k} P(Ci)\log 2(P(Ci))$$
$$(-\sum_{i=1}^{n} P(aj)\sum_{i=1}^{k} P\left(\frac{Ci}{aj}\right)\log 2(P(Ci/aj)))$$

6.2 Dataset Classification

After feature selection, the next stage is dataset classification. We carry out the classification using the optimized SVM with CSO. First, we find the best parameters of the SVM using CSO then we use the optimal parameters to build the training sample as follows.

Step 1: Define the parameters of the algorithm.

Step 2: Generate initial cats and velocity randomly.

Step 3: Distribute the cats into tracing and seeking mode.

Step 4: Check if cat is in seeking mode if yes start seeking mode otherwise start tracing mode.

Step 5: re-evaluate fitness function and keep the cat with the best solution in the memory.

Step 6: Confirm if satisfies iteration termination condition. If so, then iteration terminates and output the optimal parameter (C and 6) otherwise, return to step 2.

Step 7: Apply the optimal parameter (C, 6) and training sample to build up SVM prediction model.

6.3 Building SVM Prediction Model with Optimal Parameter (C, 6)

In this work, we use the SVM constructed by Radial basis function (RBF). RBF kernels are a family of kernels where radial function is used to smoothen a distance measure. Unlike a linear kernel, this kernel maps samples into a higher dimensional space non-linearly. The RBF kernel is represented in (2).

7. Result and Discussion

The experiment was carried on a java NetBeans platform with Weka.jar libraries to be able to access weka functionalities. First, we carry out feature selection on the 42 attributes in the NSL KDD dataset to know attributes that have high impacts and those without impact on our prediction.

After attribute selection using information gain (Entropy), some of the attributes have good entropy value while others have insignificant or zero (0) entropy value that is they have no impact on the prediction outcome. Attributes with insignificant entropy values were removed, Table 1 shows attribute with good entropy value.

7.1 Performance Evaluation

The performance of our model was evaluated based on the following metrics

S/N	Attribute Name
1	Arc_bytes
2	Dst_bytes
3	Services
4	Flag
5	Dst_host_srv_count
6	Dst_host_same_srv_rate
7	Dst_host_rerror_rate
8	Dst_host_diff_srv_rate
9	Dst_host_srv_rerror_rate
10	Diff_srv_rate
11	Count
12	Logged_in
13	Same_srv_rate
14	Rerror_rate
15	Srv_rerror_rate
16	Dst_host_srv_diff_host_rate
17	Dst_host_same_src_port_rate
18	Srv_fidd_host_rate
19	Dst_host_serror_rate
20	Dst_host_srv_serror_rate

Table 1. Attributes Selected After Information Gain

Accuracy: Proportion of the totality of correct predictions.

$$\frac{TP+TN}{P+N}$$

Precision: Proportion of correct positive observation.

$$\frac{TP}{TP+FN}$$

Recall: Proportion of positives correctly predicted as positive.

$$\frac{TP}{P}$$

F-Measure: This is derived from precision and recall values. When both precision and recall are balanced, the F-Measure produces a high result. Thus, this is very significant.

$$\frac{2*\mathrm{Re}call*\mathrm{Pr}ecision}{\mathrm{Re}call+\mathrm{Pr}ecision}$$

FP Rate: This is the ratio of mis-classified instances to the total number of normal instances. If the model generates high false alarms, its FAR value will show.

7.2 Support Vector Machine with Cat Swarm Optimization (SVM-CSO)

Prior to carrying out classification using the proposed model, it is important establishing the baseline threshold. To achieve this, the Zero R classifier provided by Weka was used. The output of the classification by this classifier is depicted in Figure 3 below which gives 54.69% as the baseline classification accuracy threshold. In Figure 4, the classification performance of the proposed model (SVM-CSO) shows that out of the 629 instances classified, 606 were correctly classified with an accuracy of 96.3%. SVM-CSO also achieved true positive (TP) Rate of 94.4%, false positive (FP) rate of 0.02, precision value of 97.5%, Recall 94.4% and F-Measure value of 96.2%. This output shows that the model performed higher than the baseline threshold.

7.3 Comparing SVOM-CSO with Some other Classifiers

Apart from establishing that the SVM-CSO performance is better than the Zero R classifier, it is important to also benchmark the performance of the model using the same dataset with some other well-known classification algorithms, namely: Table 2 and Figure 2 below both presented the brief summary of the outputs of each classification.

Figure 5 shows the classification by J48. Out of the 629

Figure 2. Comparison of Outputs

Classifier	Accurancy %	Precision %	Recall	F-Measure %	FP Rate
Zero R	54.6	29.9	54.7	38.7	0.54
CSO-SVM	96.3	95.4	97.9	96.7	0.02
J 48	95.7	96.1	94.4	95.2	0.32
Naive Bayes	84.3	76.3	94.7	843	0.24
Random Tree	95.1	95.7	93.3	94.5	0.035

Table 2. Summary of Outputs

instances classified, 602 were correctly classified with an accuracy of 95.7%. TP rate of 94.4%, FP rate of 0.032, Precision of 96.1%, Recall value of 94.4 and F-Measure of 95.2%.

Figure 6 shows the classification by Naive-Bayes, out of the 629 instances classified, 530 were correctly classified with an accuracy of 84.2%. TP rate of 94.7%, FP rate of 0.244, Precision value of 94.7%, Recall value of 94.7% and 84.5% F-Measure. Figure 7 shows the performance of Random-Tree, out of the 629 instances classified, 598 were correctly classified with an accuracy of 95.0%. TP rate of 93.3%, FP rate of 0.035, Precision value of 95.7%, Recall value of 93.3% and 94.5% F-Measure

The outputs as presented above reveal the SVM-CSO performed better than the popular classification algorithms. J48 was a classifier with closer performance but its accuracy of 95.7% and false positive rate of 0.032 compared to SVM-CSO accuracy of 96.3% and false positive rate of 0.02 showed that the optimization of SVM with CSO has helped to raise the aspiration towards an

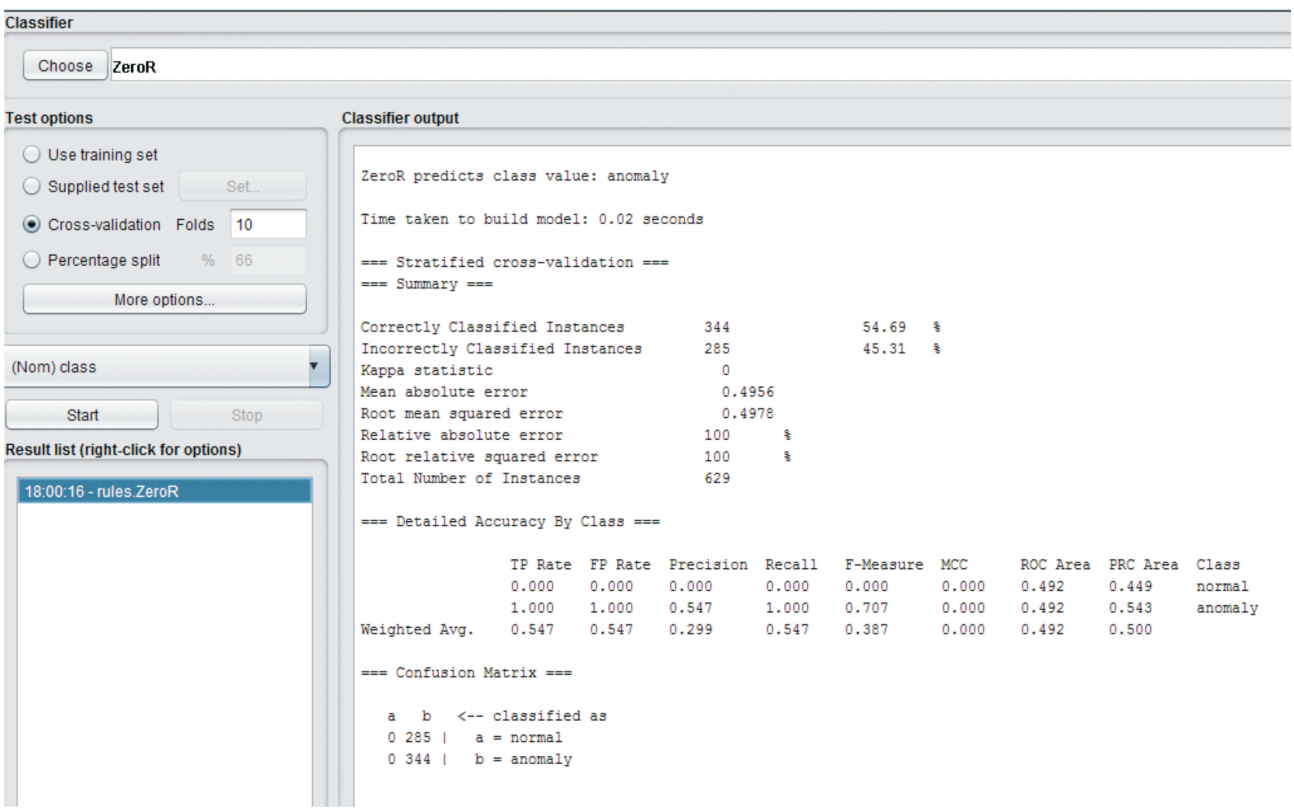

Figure 3. Zero R Classifier Output

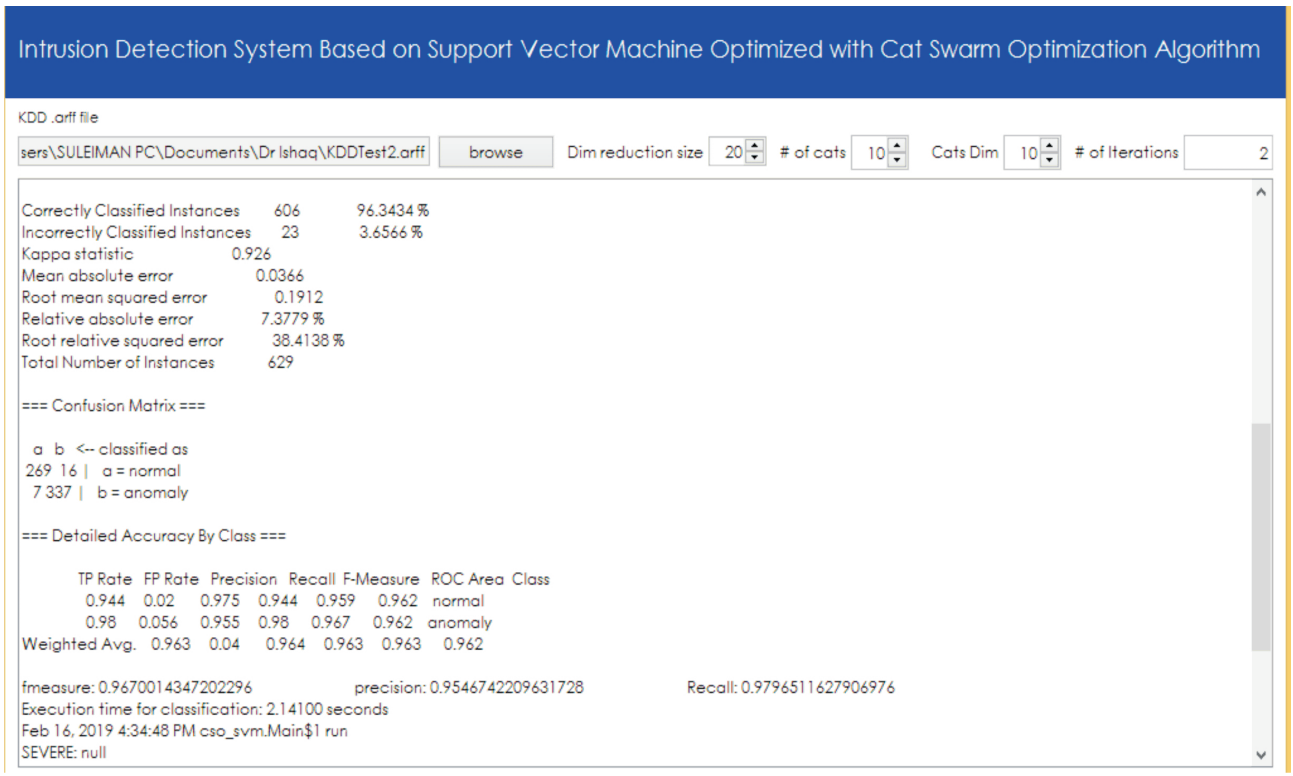

Figure 4. SVM-CSO Model Output

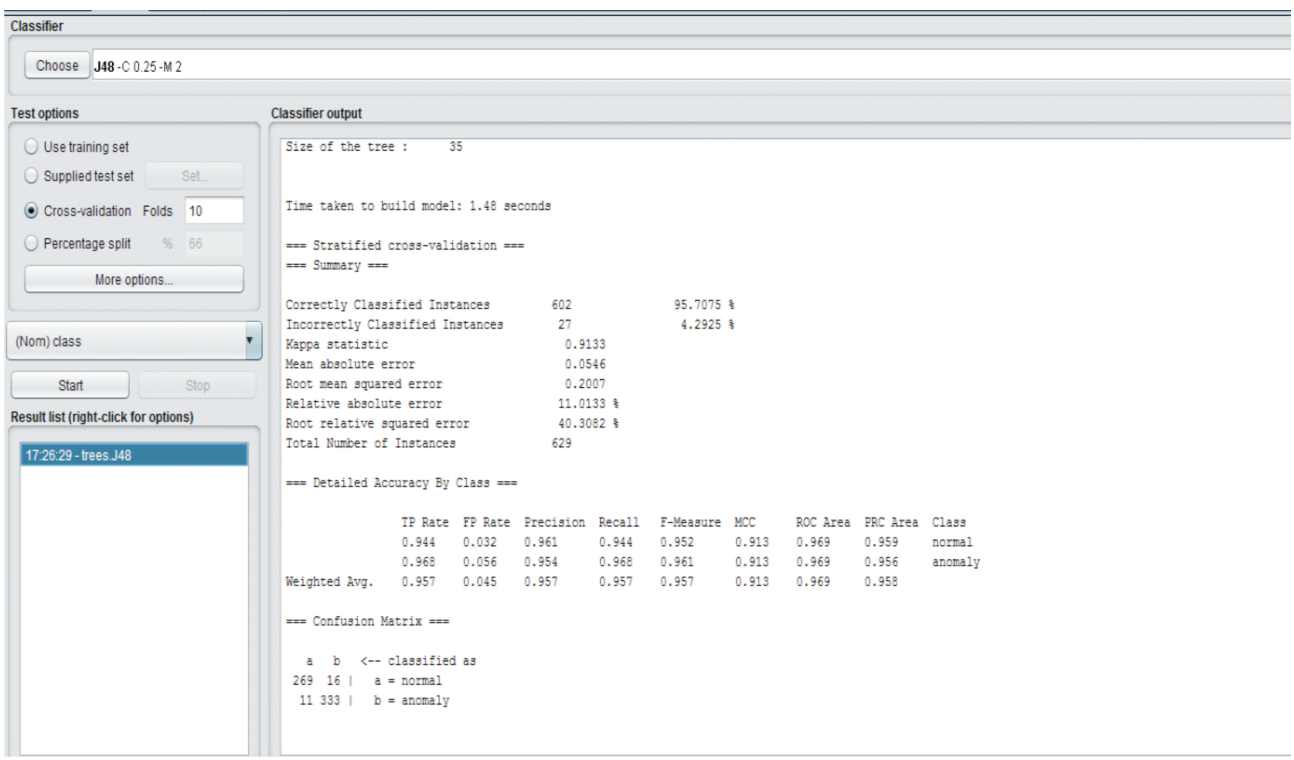

Figure 5. J48 Classifier Output

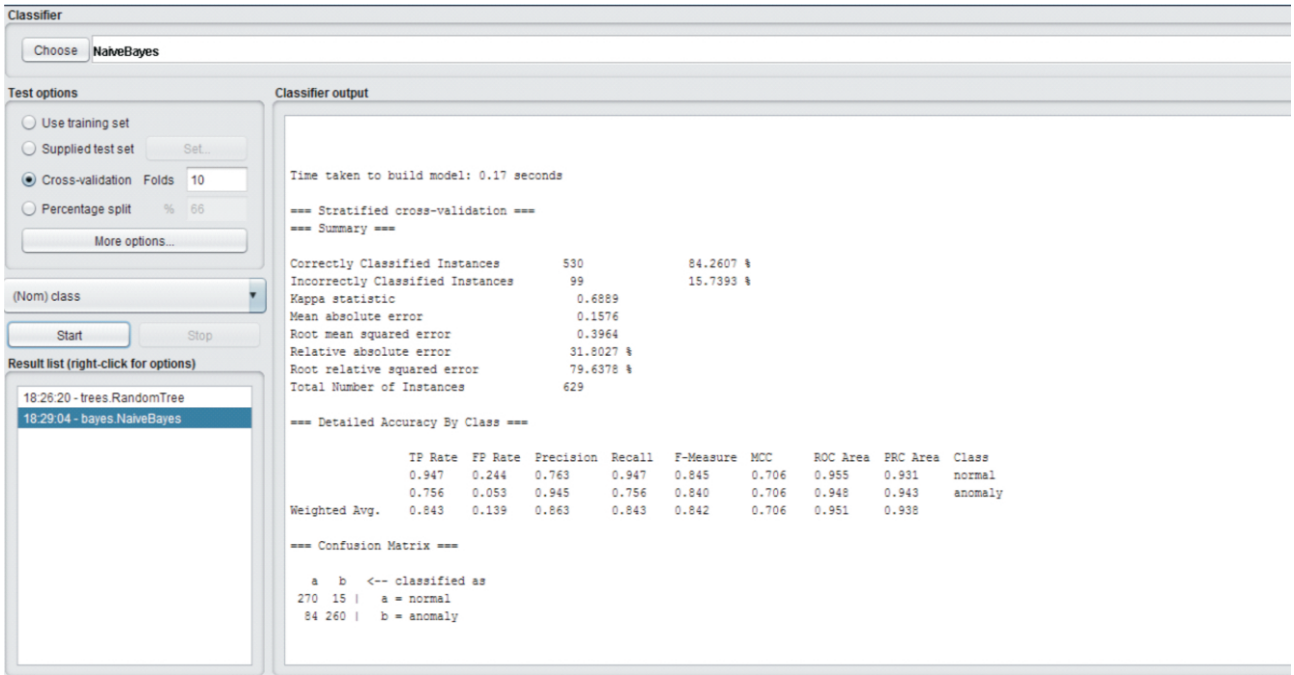

Figure 6. Naive-Bayes Output

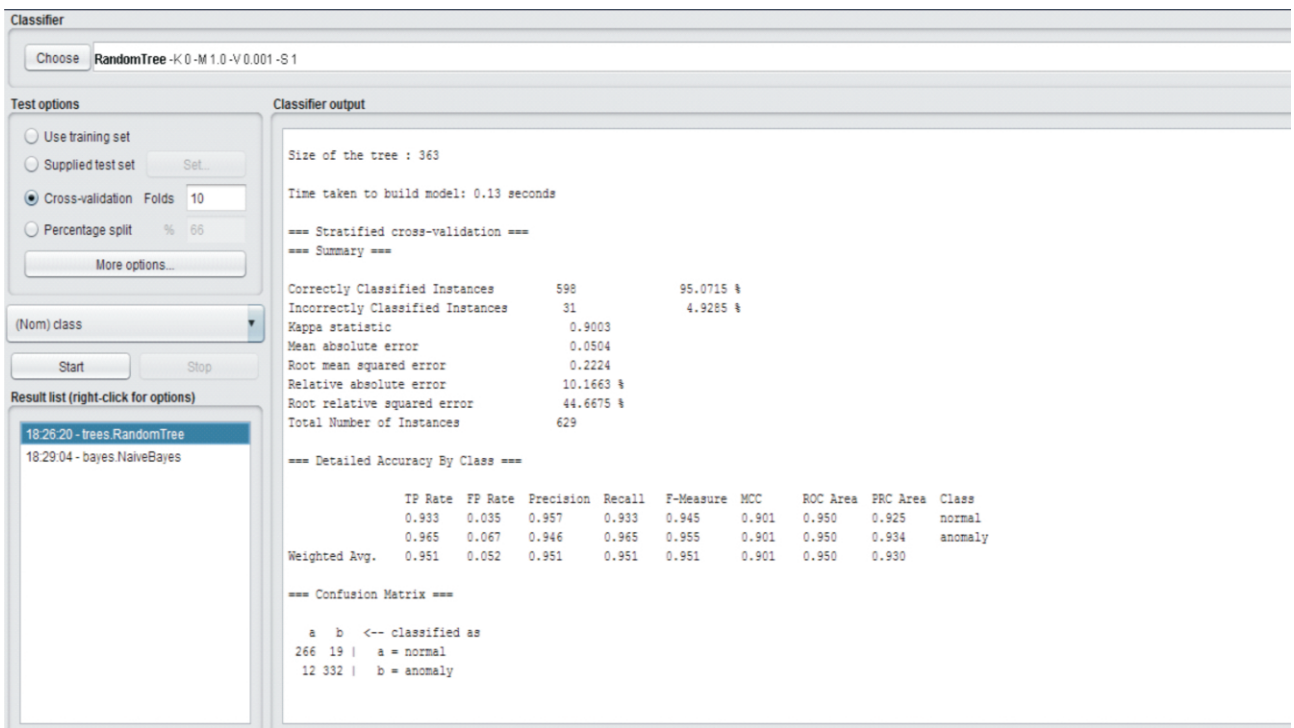

Figure 7. Random-Tree Classification Output

improved and better classification.

Conclusion

In this study, the outputs of the proposed hybridized model have been able to give an improve performance compared to other classification algorithms. With the NSL-KDD dataset, the entropy value of each of the attributes

was calculated with respect to the class value. Attributes having insignificant entropy value were removed at the preprocessing stage and classification was only carried out on the remaining attributes. Based on the classification that was done with the optimized SVM-CSO, the results achieved show that the CSO-SVM has better performance in terms of classification accuracy, false positive rate, precision and recall. In terms of accuracy which depicts the goodness in classification, and F-measure which depends on the values of precision and recall, the CSO-SVM performs better compare to other clarification algorithms like the popular J48, Naïve Bayes and Random Tree. Of importance is the low false positive rate of 0.02 of CSO-SVM compare to other hybrid algorithms such as IG-PSO-SVM and IG-ABC-SVM with 0.04 and 0.03 respectively.

References

[1]. Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science,* 25, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

[2]. Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications,* 67, 296-303. https://doi.org/10.1016/j.eswa.2016.09.041

[3]. Amami, R., Ayed, D. B., & Ellouze, N. (2015). Practical selection of SVM supervised parameters with different feature representations for vowel recognition. *International Journal of Digital Content Technology and its Applications,* 7(9), 418-424.

[4]. Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences,* 378, 484-497. https://doi.org/10.1016/j.ins.2016.04.019

[5]. Bahrami, M., Bozorg-Haddad, O., & Chu, X. (2018). Cat swarm optimization (CSO) algorithm. In *Advanced Optimization by Nature-Inspired Algorithms* (pp. 9-18). Springer, Singapore. https://doi.org/10.1007/978-981-10-5221-7_2

[6]. Daş, R., Karabade, A., & Tuna, G. (2015, May). Common network attack types and defense mechanisms. In *2015 23rd Signal Processing and Communications Applications Conference (SIU)* (pp. 2658-2661). IEEE. https://doi.org/10.1109/SIU.2015.7130435

[7]. Desai, S. P., Hadule, P. R., & Dudhgaonkar, A. A. (2017). Denial of Service Attack Defense Techniques. *International Research Journal of Engineering and Technology (IRJET),* 4(10), 1532 – 1535.

[8]. Devare, A., Shelake, M., Vahadne, V., Kamble, P., & Tamboli, B. (2016). A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis. *International Research Journal of Engineering and Technology (IRJET),* 3(04). 1917 – 1923.

[9]. Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering,* 4(6), 446-452.

[10]. Enache, A. C., & Patriciu, V. V. (2014, May). Intrusions detection based on support vector machine optimized with swarm intelligence. In *2014 IEEE 9th IEEE international symposium on applied computational intelligence and informatics (SACI)* (pp. 153-158). IEEE. https://doi.org/10.1109/SACI.2014.6840052

[11]. Hadi, I., & Sabah, M. (2015). Improvement cat swarm optimization for efficient motion estimation. *International Journal of Hybrid Information Technology,* 8(1), 279-294. https://doi.org/10.14257/ijhit.2015.8.1.25

[12]. Harshita, H. (2017). Detection and prevention of ICMP flood DDOS attack. *International Journal of New Technology and Research,* 3(3), 63-69.

[13]. Hassan, A. A., Sheta, A. F., & Wahbi, T. M. (2017). Intrusion Detection System Using Weka Data Mining Tool. *International Journal of Science and Research.* 6(9), 337 – 342.

[14]. Hoque, N., Kashyap, H., & Bhattacharyya, D. K. (2017). Real-time DDoS attack detection using FPGA. *Computer Communications,* 110, 48-58. https://doi.org/10.1016/j.comcom.2017.05.015

[15]. Kevric, J., Jukic, S., & Subasi, A. (2017). An effective

combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications,* 28(1), 1051-1058. https://doi.org/10.1007/s00521-016-2418-1

[16]. Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing,* 18, 178-184. https://doi.org/10.1016/j.asoc.2014.01.028

[17]. Kumar, A., Maurya, H. C., & Misra, R. (2013). A research paper on hybrid intrusion detection system. *International Journal of Engineering and Advanced Technology (IJEAT),* 2(4), 294-297.

[18]. Kumar, M., Mishra, S. K., & Sahu, S. S. (2016). Cat swarm optimization based functional link artificial neural network filter for Gaussian noise removal from computed tomography images. *Applied Computational Intelligence and Soft Computing,* https://doi.org/10.1155/2016/6304915

[19]. Lin, K. C., Zhang, K. Y., Huang, Y. H., Hung, J. C., & Yen, N. (2016). Feature selection based on an improved cat swarm optimization algorithm for big data classification. *The Journal of Supercomputing,* 72(8), 3210-3221. https://doi.org/10.1007/s11227-016-1631-0

[20]. Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems,* 78, 13-21. https://doi.org/10.1016/j.knosys.2015.01.009

[21]. Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks,* 13(12), 1-33. https://doi.org/10.1177%2F1550147717741463

[22]. Nidhi, M. V., & Prasad, K. M. (2016). Detection of Anomaly Based Application Layer DDoS Attacks Using Machine Learning Approaches. *i-Manager's Journal on Computer Science,* 4(2), 6-13.

[23]. Raiyn, J. (2014). A survey of cyber attack detection strategies. *International Journal of Security and Its Applications,* 8(1), 247-256. https://doi.org/10.14257/ijsia.2014.8.1.23

[24]. Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing,* 172, 385-393. https://doi.org/10.1016/j.neucom.2015.04.101

[25]. Singh, K. J., & De, T. (2017). MLP-GA based algorithm to detect application layer DDoS attack. *Journal of information security and applications,* 36, 145-153. https://doi.org/10.1016/j.jisa.2017.09.004

[26]. Tan, Z., Jamdagni, A., He, X., Nanda, P., & Liu, R. P. (2013). A system for denial-of-service attack detection based on multivariate correlation analysis. *IEEE transactions on parallel and distributed systems,* 25(2), 447-456. https://doi.org/10.1109/TPDS.2013.146

[27]. Yadav, V. K., Trivedi, M. C., & Mehtre, B. M. (2016). DDA: an approach to handle DDoS (Ping flood) attack. In *Proceedings of International Conference on ICT for Sustainable Development* (pp. 11-23). Springer, Singapore. https://doi.org/10.1007/978-981-10-0129-1_2

ABOUT THE AUTHORS

Ishaq O. Oyefolahan is an Associate Professor in the Department of Information Technology, School of ICT, Federal University of Technology, Minna, Nigeria. He holds PhD. in Information Technology. He is currently the coordinator of Applied Computing and Intelligent Systems Research Group, Deputy Dean of the School of ICT and Head of Information Technology Department. Prior to joining his current department, he was an Assistant Professor at the Department of Information Systems, International Islamic University Malaysia. His research interests are in the areas of Intelligent Systems, Business Information Systems, AI, HCI and Computing Applications Security. He has published several technical papers in international and national journals; as well as in reputable local and international conferences.

Dr. Juliana Ndunagu is a Senior Lecturer at National Open University of Nigeria, in the Faculty of Sciences and Department of Computer Sciences with Ph.D. in Information Technology (IT). In addition, Juliana Ndunagu currently serves as two term deputy dean of the Faculty.

Idris Suleiman is working as a Scientific Officer II with the National Space Research and Development Agency Abuja, Nigeria. He holds B.Tech Computer Science (Cyber Security Science) and M.Tech Computer Science from the Federal University of Technology Minna. He has enormous interest in research and development. His areas of interest are Machine learning, Computer vision, information security among others.