An Improved Anomalous Intrusion Detection Model

*[1]Bodunde O. Akinyemi, [1]Johnson B. Adekunle, [2]Temitope A. Aladesanmi,
[1]Adesola G. Aderounmu and [3]Beman H. Kamagaté

[1]Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
[2]Information Technology and Communication Unit, Obafemi Awolowo University, Ile-Ife, Nigeria
[3]Laboratoire LARIT-Abidjan Cocody Danga, Cote D'ivoire
{bakinyemi|taladesanmi|gaderoun}@oauife.edu.ng|{johnsonbayoadekunle|beman2017}@gmail.com

Abstract - The volume of cyber-attack targeting network resources within the cyberspace is steadily increasing and evolving. Network intrusions compromise the confidentiality, integrity or availability of network resources causing reputational damage and the consequential financial loss. One of the key cyber-defense tools against these attacks is the Intrusion Detection System. Existing anomalous intrusion detection models often misclassified normal network traffics as attacks while minority attacks go undetected due to an extreme imbalance in network traffic data. This leads to a high false positive and low detection rate. This study focused on improving the detection accuracy by addressing the class imbalanced problem which is often associated with network traffic dataset. Live network traffic packets were collected within the test case environment with Wireshark during normal network activities, Syncflood attack, slowhttppost attack and exploitation of known vulnerabilities on a targeted machine. Fifty-two features including forty-two features similar to Knowledge Discovery in Database (KDD '99) intrusion detection dataset were extracted from the packet meta-data using Spleen tool. The features were normalized with min-max normalization algorithm and Information Gain algorithm was used to select the best discriminatory features from the feature space. An anomalous intrusion detection model was formulated by a cascade of k-means clustering algorithm and random-forest classifier. The proposed model was simulated and its performance was evaluated using detection accuracy, sensitivity, and specificity as metrics. The result of the evaluation showed 10% higher detection accuracy, 29% sensitivity, and 0.2% specificity than the existing model.

Keywords— anomalous, cyber-attack, Detection, Intrusion

1 Introduction

The digital age has influenced the way business is transacted and government operations are carried out. These activities now rely on an interdependent network of information technology infrastructures called cyberspace. Cyberspace is the collection of computing devices connected by networks in which electronic information is stored and utilized, and communication takes place (Hughes, 2010). Cyberspace also called cyber environment (Phister, 2010), includes users, networks, devices, all software, processes, information in storage or transit, applications, services, and systems that can be connected directly or indirectly to networks. Cyberspace allows the interdependent network of information technology infrastructures, telecommunications networks, such as the Internet, computer systems, integrated sensors, system control networks and embedded processors and controllers common to global control and communications.

As the systems that support human daily life such as security, healthcare, electricity, financial services, transportation and communication services become increasingly interconnected in the cyberspace, the volume of the targeted attack is steadily increasing and evolving forcing businesses to revamp their network security systems to avert network intrusions and the consequential financial loss. It seems "every organization is at the risk of a cyber-attack" (Arabo, 2015). According to a recent survey, attacks within cyberspace has risen to 80 to 90 million plus per year and more than 50 percent of cyber intrusions on companies, corporations, government and campus networks are going undetected (Kirk, 2015).

Existing anomalous intrusion detection models often misclassified normal network traffics as attacks while minority attacks go undetected due to extreme an imbalance in network traffic data. This leads to a high false positive and low detection rate. Therefore, an effort is made in this work to develop an improved anomalous intrusion detection model that would effectively detect both majority and minority attacks based on anomalous network traffic in cyberspace. The rest of this paper is arranged as follows: Section 2 discusses the related works, Section 3 described the modelling process while Section 4 described the prediction algorithm and the conclusions are discussed in Section 5.

2 Related Works

Several algorithms have been developed for intrusion detection using the traditional solutions, but traditional Intrusion Detection System (IDS) algorithms are not optimized for public cloud environments. A traditional IDS inspects network traffic looking for known attack signatures and either alerts on the traffic or stops it from proceeding into the network, depending on how it has been deployed. However, the current state of the art seeks to further provide extensive coverage of network protocols to detect a wider range of attacks. In a cloud environment, malicious attacks and intrusions are dynamic and are needed to perform intrusion detection in a real-time environment with data streams, thus, Data mining methods such as clustering, classification, and association rule mining are often used to obtain valuable information of network intrusion through analysing the network data.

Minnesota Intrusion Detection System (MINDS) developed by Ertoz *et al.*, (2004), efficiently used data mining techniques for anomaly detection. It contains two major modules namely the anomaly detection module and association pattern analysis. The anomaly detection module uses Local Outlier Factor (LOF) to detect

*Corresponding Author

anomalies and assigns a score to each data point based on the factor. A human analyst then verifies whether the data point is a real intrusion or normal behaviour. Association pattern analysis is used to summarize the anomalous network connections. The major disadvantage of MINDS system is that it requires a human analyst to verify network connections.

In Gong *et al.,* (2005), Genetic algorithm was used to detect intrusion in networks through effective feature selection. Features were extracted using information gain to reduce the dataset dimensionality. Then, they formed a linear structure rule from the selected features in order to classify network behaviours into normal and anomalous behaviour. However, their approach considers only discrete features. In Wang *et al.,* (2010), campus network security problems were analyzed. An intrusion detection structure that can detect attacks using a data mining algorithm was designed. Associated rules were created in intrusion detection to find association relation in the network data stream in the algorithm. The system could detect the intrusion attack behaviour in the campus network, especially the internal attacks on a campus network. It was, however a host-based intrusions detection system and limited its data set to only applications on the network without considering other features and elements on the network.

In Winter *et al.*, (2011), the inductive network intrusion detection system was proposed. The system operates on lightweight network flows and uses One-Class Support Vector Machines (OCSVM) for analysis. But the system was trained with malicious rather than with normal network data. Evaluations brought satisfying results. They achieved a 0% false alarm with detection rate around 98%. The drawbacks of this work that the attack variations are unlimited, this leads to having big differences in class density which affect the detection performance of the OCSVM. Also, it is impossible to have a representative dataset of all possible attacks that could happen in the future.

A new Real Time Unsupervised Network Intrusion Detection System (NIDS) which monitor network flows in two windows with different sizes and detect network attacks by correlating outliers from multiple clusters was proposed in Amoli, & Hamalainen, (2013). The proposed NIDS has the ability to detect different types of intrusions in real-time such as DOS, DDOS, scanning, distribution of worms and any other network attacks which produce a huge amount of network traffic and in the meanwhile, it detects Bot-Master if the detected attack launched by Bots. The authors didn't mention how to distinguish between normal and abnormal packets, also the limitation of this approach is the use of DBSCAN algorithm which fails when the density is varied in normal instances. Also, it can't detect attacks other than flood attacks.

In Almutairi, and Parish (2014), a predictive intrusion detection model that is based on usage of classification techniques such as decision tree and Bayesian techniques was proposed. The model was trained using KDD"99 intrusion detection dataset. The results showed that the decision tree algorithm J48 based on C4.5 provides 99.95% of correctly classified instances and was better than the Naïve Bayes technique. It was also found that false positives using Naïve Bayes were high for Probing and Remote to Local attack categories. The shortcoming of their model is the inability to detect novel attacks with a different distribution than those attacks in the training dataset based on classifier used. The feature selection also may eliminate some features which are relevant to future attacks. Also, it is a problem to have labelled data in real environment. A software (Spleen) to obtain KDD compatible features from live network traffic packets was proposed by Guillén *et al.,* (2012), with the possibility of getting additional features information in order to construct an updated data set with multiple networking scenarios. The work established the usefulness of the additional features in improved detection of remote to local attack.

In Hussein, (2014), a Very Fast Decision Tree (VFDT) classification was used to build an anomalous intrusion detection system. VFDT is a high-performance data mining system based on decision trees. It is fast and the description of classifiers that it derives is easily understood. VFDT deals with data streams. As data arrives, this data stream grows gradually while the data is classified (Nguyen and Choi, 2008). VFDT does not accumulate the examples in main memory, because it can gradually grow without waiting for the arrival of all the examples. The VFDT gradually grows as examples are received to create leaf nodes, which grow into branches from only the root node. In Maglaras *et al.*, (2016), ensemble methods and social network metrics for improving the accuracy of once class Support Vector Machine for intrusion detection system in SCADA system was proposed. The authors modelled interaction among network nodes using Social Network Analysis (SNA). The assumption is that those network anomalies are the patterns of interaction that significantly differ from the normal interaction. Detection accuracy was improved than previous models but at a very high computational and time complexity. In this paper, an attempt was made to develop an extensive algorithm that would alleviate false alarm rates due to extreme imbalance in network traffic data.

3 Methodology

The conceptual model description and the detailed algorithm of the proposed model formulation are presented as follows.

3.1 Proposed Model Architecture

In this study, the model of Hussein, (2014) shown in Fig 1 was improved. In this existing system, Network packets were captured; basic features were extracted from the network packet headers and the content payloads. Features with high information gain were selected, normalized and converted into linear discrete integer values in order to avoid the impact of overpowering the large-scale features on the other features. The pre-processed data were fed into the VFDT classification algorithm to classify the traffic as normal or attack.

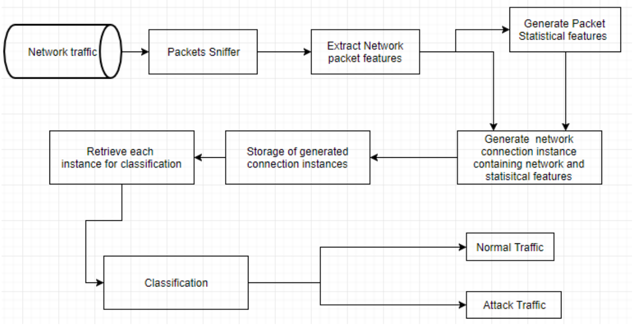

Network traffic
Packets Sniffer
Extract Network packet features
Generate Packet Statistical features
Generate network connection instance containing network and statisitcal features
Storage of generated connection instances
Retrieve each instance for classification
Classification
Normal Traffic
Attack Traffic

Fig. 1: Existing Model Overview (Hussein, 2014)

The proposed model flow is as shown in Figs 2 and 3. It follows a sequential architecture from the extraction of network traffic packet features to classification of network traffic connection as normal or attack. In the proposed model, network traffic packets are clustered before applying a classification algorithm on each of the clusters. The aim of this enhancement is to increase detection accuracy, especially on non-dominant network attack traffics.

The proposed model uses a cascade of two machine learning algorithm: K-means, an unsupervised machine learning algorithm and Random Forest, a supervised machine learning algorithm. The detection was in two stages. In the first stage, k-Means clustering algorithm was applied to the training dataset to obtain k disjoint clusters based on the attributes. The packets in the same cluster are similar in term of Euclidean distance measurement between the instances and their cluster centroids. Cascading the decisions from the k-Means and Random Forest classifier was done in two phases: 1) the Selection phase, and 2) the Classification phase. In the selection phase, the closest cluster i.e., the nearest neighbour cluster to the test instance was selected. In the selected cluster the Random Forest classifier corresponding to that cluster was generated. In the Classification phase, the test instance was classified into normal or anomaly using the Random Forest classifier result.

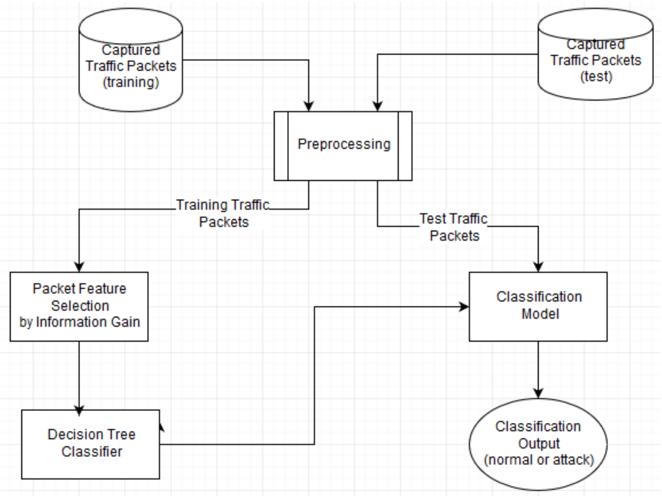

Captured Traffic Packets (training)
Captured Traffic Packets (test)
Preprocessing
Training Traffic Packets
Test Traffic Packets
Packet Feature Selection by Information Gain
Classification Model
Decision Tree Classifier
Classification Output (normal or attack)

Fig. 2: Proposed Model flow

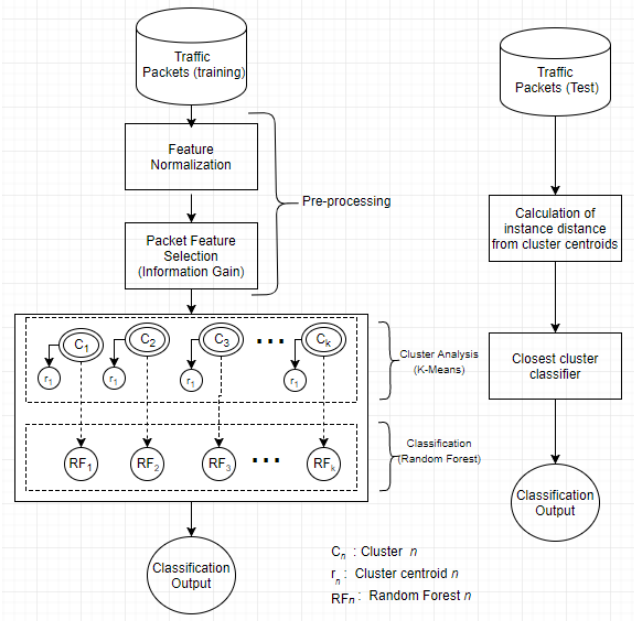

Traffic Packets (training)
Traffic Packets (Test)
Feature Normalization
Packet Feature Selection (Information Gain)
Pre-processing
Calculation of instance distance from cluster centroids
C1
C2
C3
Ck
r1
Cluster Analysis (K-Means)
Closest cluster classifier
RF1
RF2
RF3
RFk
Classification (Random Forest)
Classification Output
Classification Output
Cn : Cluster n
rn : Cluster centroid n
RFn : Random Forest n

Fig. 3: Proposed Model details

3.2 Model Formulation

The proposed model formulation is detailed in Algorithm1. The model starts from the identification of network domain within the cyberspace and selected hosts or gateways to be monitored by the intrusion detection system. Fig 4 shows the entire process in a block diagram.

3.3 Stages of Model Development

The processes involved in model development are as follows:

3.3.1 Data Collection

The network traffic training dataset used in this study were collected in the Cyber Security Laboratory environment within OAUnet.

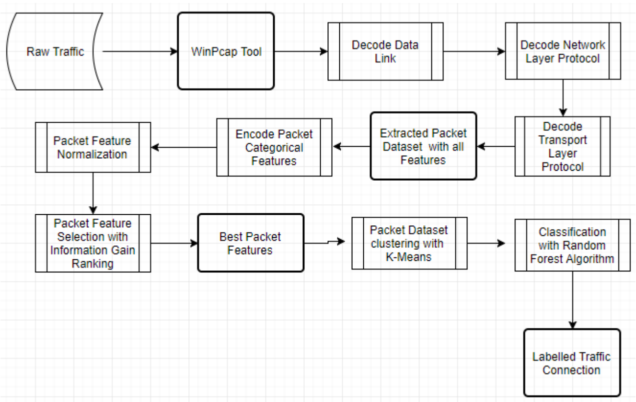

Raw Traffic
WinPcap Tool
Decode Data Link
Decode Network Layer Protocol
Decode Transport Layer Protocol
Extracted Packet Dataset with all Features
Encode Packet Categorical Features
Packet Feature Normalization
Packet Feature Selection with Information Gain Ranking
Best Packet Features
Packet Dataset clustering with K-Means
Classification with Random Forest Algorithm
Labelled Traffic Connection

Fig 4: Block diagram depicting the entire classification process

i. Identify a network environment within the cyberspace (problem domain)
ii. Identify one or more hosts/gateways to be monitored for cyber intrusion
iii. Monitor, log and clearly label inbound and outbound network traffic on the network interfaces using network sniffer(packet capture)
iv. Extract and aggregate packet headers, protocol, services, source port, destination port and other attributes using Spleen tool.
v. Encode categorical attributes into numeric attributes using the Categorical attribute encoding algorithm.
vi. Export/Save Spleen report as .arff (attribute relation file format) file.
vii. Normalized each packet attributes using min-max normalization :

$$Q = \left(\frac{P - Min(P)}{Max(P) - Min(P)}\right) * (N - M) + M$$

where value of feature P needs to be normalized into value Q. Min(P) and Max(P) represents minimum and maximum values in the feature P respectively. M and N represents the Lower value and upper Value respectively in the new range. Q is set to zero if the maximum is equal to the minimum.
viii. Calculate the relevance of each feature (Information Gain) to the classification
ix. Select optimal feature subset from the feature space using SelectOptimalfeature Algorithm
x. Divide the dataset into training and testing subset in ratio 7:3, 70% for training an 30% for testing the model.
(a) Cluster the training dataset by applying k-Means clustering algorithm
(b) For each cluster in (a), apply Random forest classification algorithm on each cluster instance of training data using Random Forest classification algorithm.
xi. For each instance Zi in testing data
(a) Calculate the instance distance from the centroid of each training data cluster using euclidean distance measurement.

$$D(Z_i, r_j) = \sqrt{\sum_{k=1}^{p} (Z_{ik} - r_{jk})^2}$$

Where D(Zi,rj) is the distance between instance Zi and cluster Zik is attribute k of instance Zi, rjk is attribute k of centroid rj and p is the number of attributes extracted from the packet
(b) Classify the instance using the closest cluster rj to instance Zi
Iterate through the dataset until all instances have been classified.

Algorithm 1. Proposed model formulation

Real normal traffic is not always guaranteed to be free from intrusion traces, however, data collection from a controlled environment gave us higher percentage guarantee that inbound and outbound traffics are intrusion-free than the uncontrolled environment. The controlled environment also provides the opportunity to execute real life attack script against some targeted machine which will be absolutely illegal if run against production machine. The environment also makes it easy to filter out unwanted packets during network attack traffic collection.

Fig 5 shows the Cyber Security Laboratory network and server environment. The web server hosts all web applications being developed in the laboratory with support for Apache server and Windows server running in Vmware virtual machine. All security tools including Kali Linux and Metasploitable (Beggs, 2014) were installed on a security application server in a virtualized environment. Normal network traffics were collected at random period within 24hours of the day for a whole week. All the network packets collected within the period were pre-processed and labelled as normal traffic for model training. Attack traffics were also collected at a specific time and day during the process. Attack scripts gathered from related works and certified security specialists for performing reconnaissance, root access attempts, denial of service attacks, probe attacks etc. were executed to exploit vulnerabilities in targeted nodes within the lab. The IP address of attack source and destination were noted during the process in other to filter out non-attack traffic from other nodes. The network traffic generated during this process were collected, pre-processed and labelled as attack traffic for model training.

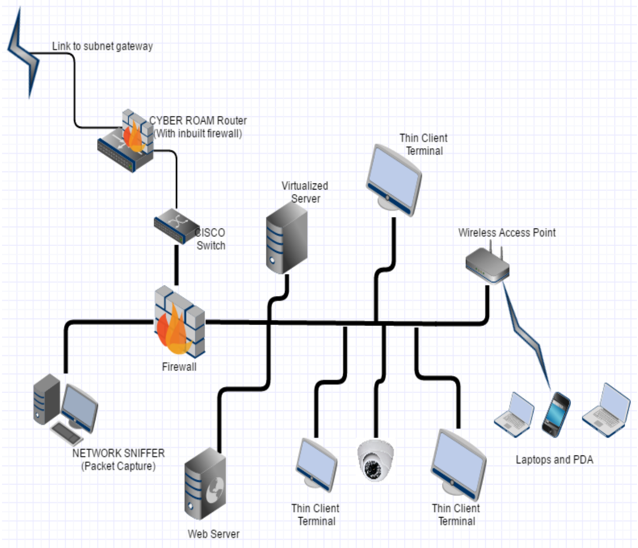


Fig. 5: Network description of data collection environment

3.3.2 Feature Extraction

Fifty-two features were extracted from collected traffic packets using Spleen. Features extracted include basic and host-based features from packet headers, time-based features from traffic connection and content-based features from packet content and connection status.

- Basic: These features corresponds to fields in the network packet headers and session timeouts. These are useful for detecting attacks which target protocol and service vulnerabilities.
- Time: These are important for identifying high volume fast rate DoS attacks based on the number.
- Host: These features store the number of connections to the same host, port or service by a destination host in the last 100 connections. They are useful for identifying probe attacks.
- Content: These are based on domain knowledge. Important for detecting stealthy attacks (U2R, R2L) by observing the payload section of the packets.

For improved classification accuracy, the attributes were normalized using of connections requests to the same destination host or service in a very short time frame min-max normalization process. Normalization ensures that equal weight is given to all attributes.

3.3.3 Feature Reduction

Feature reduction process is used to reduce the original feature space and select the most important features for the classification. Information Gain criteria is used for feature selection in this study for feature selection. An entropy value of each attribute of dataset instance is calculated, the attributes are then ranked in descending order of information gain. Optimal features were selected with Algorithm 2. Fifty-three (53) network traffic features were originally extracted including the class label. Information Gain algorithm was applied on the dataset to rank the features according to their effectiveness in distinguishing normal classes from abnormal classes. All features having information gain less not less than 0.0000249, 0.000167, 0.00000332 and 0.000000282 were selected for DoS, probe, user-to-root and remote-to-local attacks respectively; other features were ignored because of negligible information gain.

```
SelectOptimalFeature (Dataset, Information gain     threshold)
INPUT: X (dataset), T (Information gain threshold)
OUTPUT: Topmost n attributes according to
information gain weight
BEGIN
    Sort features by descending order of Information Gain
                $TotalIG = \sum_{i}^{N} IG(C;x_i)$
    for each feature xi  in dataset X
                $InformationGainWeight(W_{xi}) = \frac{IG(C;xi)}{TotalIG}$
    end
    Select minimum n top features, where the sum of the
weights      of these features is greater than the Information
Gain         threshold
                    i =1
$W_n = W_n + W(x_i)$
                $If\ W_n \geq T$
            GOTO Step 13
        else
            Increment I and GOTO step 8
    Select top n features as new feature set
  END
```

Algorithm 2: Optimal Features Subset Selection

3.3.4 Classification

The classification was done by a cascade of K-means clustering algorithm and Random forest classifier. The cascade of K-means algorithm and Random Forest classifier was used to overcome the class imbalance problem which is prevalent in intrusion detection data. The detection is in two stages. In the first stage, k-Means clustering algorithm is applied to the training dataset to obtain k disjoint clusters based on their attributes. The packets in the same cluster are similar in term of Euclidean distance measurement between the instances and their cluster centroids.

The dataset was divided into training and testing subset in ratio 7:3, 70% for training and 30% for testing the model as widely used in literatures (Hussein, 2014). K-means clustering algorithm was applied to the training data set to obtain k disjoint clusters of the traffic connection. Each k-Means cluster represents a region of similar traffic connection in terms of Euclidean distances between the instances and their cluster centroids. For each k-means cluster, Random Forest classification algorithm was applied on each k-means cluster. During testing phase, the distance between each testing instance and the centroid of each k-means cluster are calculated. The Random forest tree of the closest instance is then used to classify the instance as a normal or attack instances.

3.4 Simulation

The simulation was done in WEKA knowledge flow environment. The network traffic dataset collected were divided into training and test dataset in ratio 70 to 30 percent respectively. Tables 2 and 3 present the statistics of training and test dataset and the number of traffic type sessions present for each group of intrusion. The two attribute relation format file loaders (*arff l*oader) were used to load the training and test dataset into the model. “Class assigners” and “Dataset Maker (training and test)” were used to indicate the class label and distinguish between training dataset and test dataset. K-means clustering algorithm and Random Forest classification algorithm were cascaded as a filtered classifier. The datasets were fed into the existing model and the performance evaluation parameters were recorded.

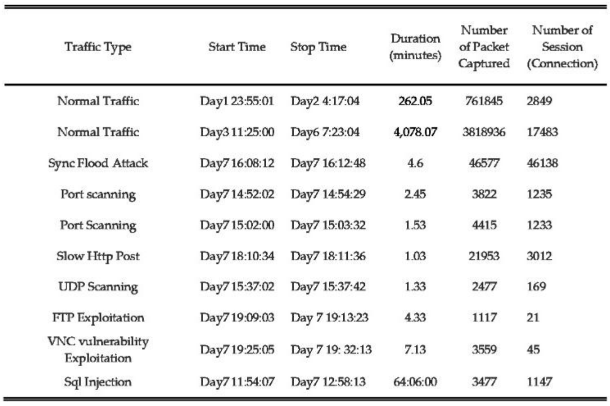
4 Results

In this section, the results of the study are presented as follows.

4.1 Data Collection Results

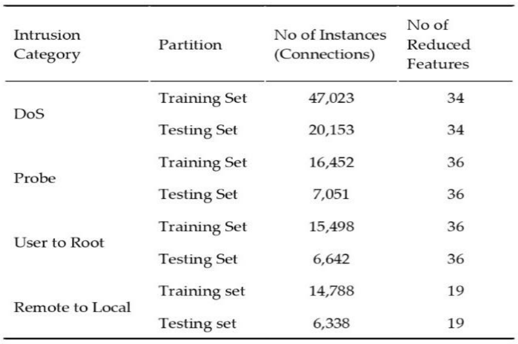
Table 1 presents the statistics of traffic collected within a one-week period. Normal network traffics were for 6 days were abnormal traffics were generated, collected, and random interval on the 7th day. The data confirmed the massive generation of the massive network packet in cases of DoS attacks like *Sync flood* and *slowhttpost*. Remote to local and user to root attacks generated the least amount of packets despite being the deadliest attacks of the four categories, hence the reason for the difficulty in detecting them.

Table 1. Network Traffic Type and Their Capture Periods (Time in 24 Hour Format)



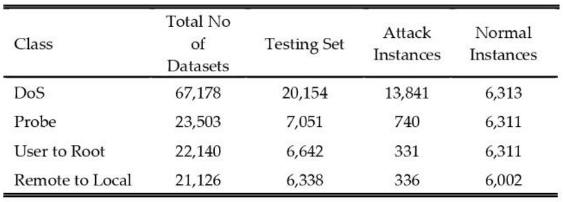
Traffic Type	Start Time	Stop Time	Duration (minutes)	Number of Packet Captured	Number of Session (Connection)
Normal Traffic	Day1 23:55:01	Day2 4:17:04	262.05	761845	2849
Normal Traffic	Day3 11:25:00	Day6 7:23:04	4,078.07	3818936	17483
Sync Flood Attack	Day7 16:08:12	Day7 16:12:48	4.6	46577	46138
Port scanning	Day7 14:52:02	Day7 14:54:29	2.45	3822	1235
Port Scanning	Day7 15:02:00	Day7 15:03:32	1.53	4415	1233
Slow Http Post	Day7 18:10:34	Day7 18:11:36	1.03	21953	3012
UDP Scanning	Day7 15:37:02	Day7 15:37:42	1.33	2477	169
FTP Exploitation	Day7 19:09:03	Day 7 19:13:23	4.33	1117	21
VNC vulnerability Exploitation	Day7 19:25:05	Day 7 19: 32:13	7.13	3559	45
Sql Injection	Day7 11:54:07	Day7 12:58:13	64:06:00	3477	1147

Table 2. Partitioning of Network Traffic Dataset for Simulation



Intrusion Category	Partition	No of Instances (Connections)	No of Reduced Features
DoS	Training Set	47,023	34
	Testing Set	20,153	34
Probe	Training Set	16,452	36
	Testing Set	7,051	36
User to Root	Training Set	15,498	36
	Testing Set	6,642	36
Remote to Local	Training set	14,788	19
	Testing set	6,338	19

Table 3. Test Dataset Statistics



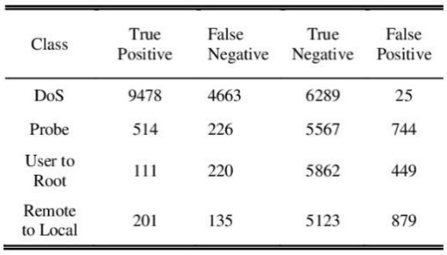
Class	Total No of Datasets	Testing Set	Attack Instances	Normal Instances
DoS	67,178	20,154	13,841	6,313
Probe	23,503	7,051	740	6,311
User to Root	22,140	6,642	331	6,311
Remote to Local	21,126	6,338	336	6,002

4.2 Model Evaluation Result

Tables 4 and 5 presents the simulation results of existing and proposed model for each type of attack detailing the number of true positives, true negatives, false positive and false negatives. The model performance is evaluated using detection accuracy, sensitivity and specificity as metrics. The evaluation metrics are basically derived from the four basic attributes of the confusion matrix depicting the actual and predicted classes. These parameters were used to derive the accuracy, sensitivity and specificity of the existing model (Hussein, 2014) and the proposed model as shown in Tables 6 and 7.

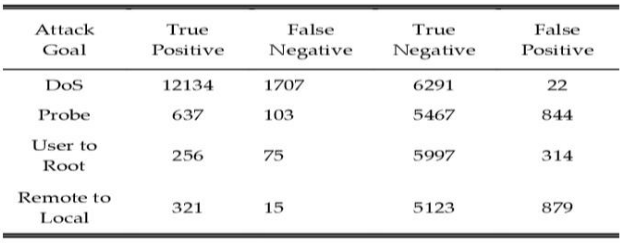
Figs 6, 7 and 8 present a comparison of accuracy, sensitivity and specificity of the existing and the proposed anomalous intrusion detection model. Fig 9 shows the Receiver Operating Curve of the model with 0.8521969 Area Under the Curve (ROC_AUC). Since the curve drawn closer to the left border of the graph, then it denotes a higher accuracy.

Table 4. Simulation Results of the Existing Model



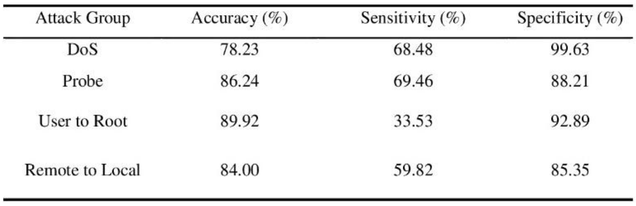
Class	True Positive	False Negative	True Negative	False Positive
DoS	9478	4663	6289	25
Probe	514	226	5567	744
User to Root	111	220	5862	449
Remote to Local	201	135	5123	879

Table 5. Simulation Results of the Proposed Model



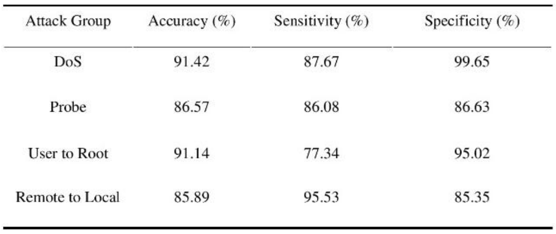
Attack Goal	True Positive	False Negative	True Negative	False Positive
DoS	12134	1707	6291	22
Probe	637	103	5467	844
User to Root	256	75	5997	314
Remote to Local	321	15	5123	879

Table 6. Evaluation Results of the Existing Model



Attack Group	Accuracy (%)	Sensitivity (%)	Specificity (%)
DoS	78.23	68.48	99.63
Probe	86.24	69.46	88.21
User to Root	89.92	33.53	92.89
Remote to Local	84.00	59.82	85.35

Table 7. Evaluation Results of the Proposed Model



Attack Group	Accuracy (%)	Sensitivity (%)	Specificity (%)
DoS	91.42	87.67	99.65
Probe	86.57	86.08	86.63
User to Root	91.14	77.34	95.02
Remote to Local	85.89	95.53	85.35

It was noted that the proposed model has a detection accuracy of approximately 10% higher when compared with the detection accuracy of the existing model. This implies that the proposed model could detect more network intrusion than the existing model. Similarly, the proposed model has higher sensitivity in all attack groups. This implies that the proposed model is more sensitive in detecting intrusion than the existing model. This reduces the false acceptance of an intrusion network session as a normal session. Also, the proposed model has 2.16 % higher specificity for UserToRoot attack category and slightly higher (0.02%) specificity for DoS attack group. This implied a decrease in the false alarm rate in the proposed model by approximately 2%.

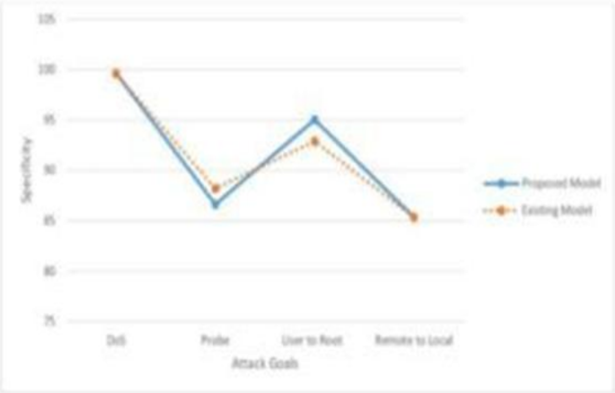


Fig. 6: Comparison of the Accuracy for the Proposed and Existing Models

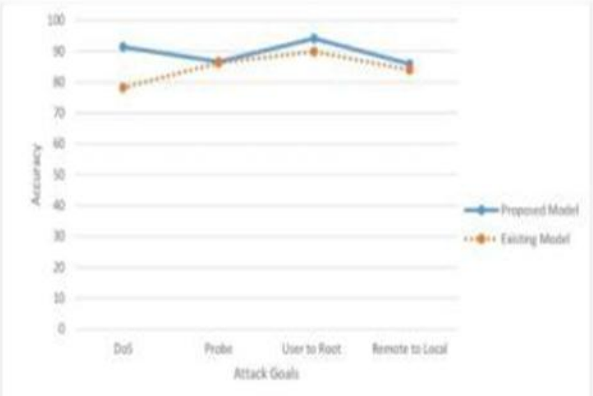


Fig. 7: Comparison of the Specificity for the Proposed and Existing Models

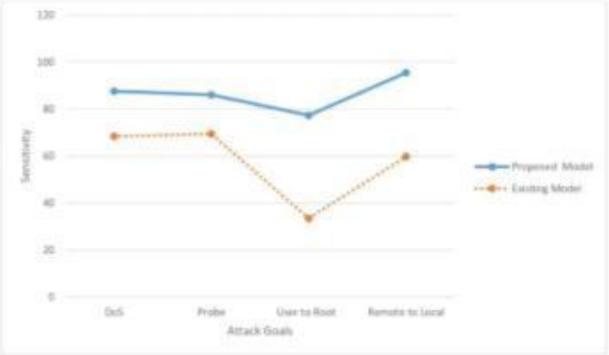


Fig. 8: Comparisons of the Sensitivity for the Proposed and Existing Models

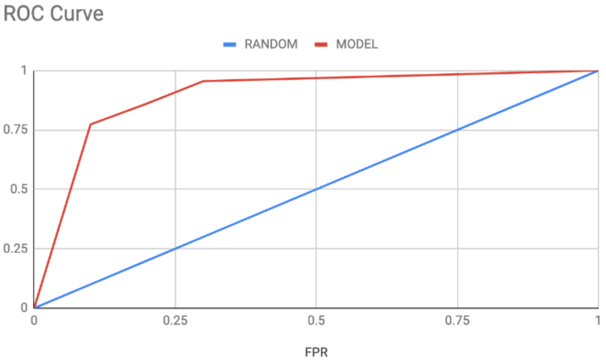


Fig. 9: Receiver Operating characteristic (ROC) curve for the model

5 Conclusion

An anomalous intrusion detection model was developed in this study to increase detection accuracy. The proposed model has a higher detection accuracy of 91.42%, 86.57%, 94.14% and 85.89% for DoS, Probe, UserToRoot, and RemoteToLocal intrusion respectively when compared with the detection accuracy of the existing model which are 78.23%, 86.24%, 89.92% and 84.00% for DoS, Probe, UserToRoot, and RemoteToLocal intrusion respectively. The study concluded that the developed model proved to have higher detection accuracy, sensitivity and specificity than the existing model and could be implemented to solve the identified problems cyberspace intrusion detection.

Further work could be done by expanding the scope of attacks for generating abnormal network traffic to include advanced persistent threat which takes months or possibly years to materialize. This will require monitoring and collecting the network traffic dataset for a longer period, running to months and years. Other open areas of work include deep inspection of packet payload content in addition to packet meta-data extracted in this work. Other feature reduction algorithms could be exploited aside information gain to select optimal features for the classification model

6 Acknowledgment

This Research was funded by the TETFund Research Fund and Africa Centre of Excellence OAK-Park, Obafemi Awolowo University, Ile-Ife.

References

Arabo, A. (2015). Cyber security challenges within the connected home ecosystem futures. *Procedia Computer Science*, *61*, 227-232.

Almutairi, A., & Parish, D. (2014, December). Using classification techniques for creation of predictive intrusion detection model. In IEEE *9th International Conference for Internet Technology and Secured Transactions (ICITST)*, 223-228

Arabo, A. (2015). Cyber security challenges within the connected home ecosystem futures. *Procedia Computer Science*, *61*, 227-232.

Amoli, P. V., & Hamalainen, T. (2013, October). A real time unsupervised NIDS for detecting unknown and encrypted network attacks in high speed network. In *2013 IEEE International Workshop on Measurements and Networking Proceedings (M&N),* 149-154.

Beggs, R. W. (2014). *Mastering Kali Linux for Advanced Penetration Testing*. Packt Publishing Ltd.

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., & Dokas, P. (2004). Minds-minnesota intrusion detection system. *Next generation data mining*, 199- 218.

Gong, R. H., Zulkernine, M., & Abolmaesumi, P. (2005, May). A software implementation of a genetic algorithm based approach to network intrusion detection. In IEEE *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing,2005 and First ACIS International Workshop on Self- Assembling Wireless Networks. SNPD/SAWN 2005.* 246-253.

Guillén, E., Rodriguez, J., Páez, R., & Rodriguez, A. (2012, October). Detection of non-content based attacks using GA with extended KDD features. In *Proceedings of the world congress on engineering and computer science*, 30-35.

Hughes, R. (2010). A treaty for cyberspace. *International Affairs*, *86*(2), 523-541.

Hussein, N. A. (2014). *Design of a Network-Based Anomaly Detection System Using VFDT Algorithm* (Doctoral dissertation, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ)).

Kirk, R. (2015). Threat Sharing–A Neighbourhood Watch For Security Practitioners. *Network Security*, *2015*(12), 5-7

Maglaras, L. A., Jiang, J., & Cruz, T. J. (2016). Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. *Journal of Information Security and Applications*, *30*, 15-26.

Nguyen, H. A., & Choi, D. (2008, October). Application of data mining to network intrusion detection: classifier selection model. In *Asia-Pacific Network Operations and Management Symposium,* Springer Berlin Heidelberg. 399-408.

Phister Jr, P. W. (2010). *Cyberspace: The ultimate complex adaptive system*. Air Force Research Lab Rome NY.

Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Systems with Applications*, *37*(9), 6225-6232.

Winter, P., Hermann, E., & Zeilinger, M. (2011, February). Inductive intrusion detection in flow-based network data using one-class support vector machines. In *International Conference on New Technologies, Mobility and Security*. 1-5.