REDUCING THE OBSERVABLE STATES SPACE OF HIDDEN MARKOV MODEL FOR DISTRIBUTED DENIAL OF SERVICE ATTACK PREDICTION USING KULLBACK-LIEBLER DIVERGENCE

[1,2]Afolorunso*, A. A., [1]Adewole, A. P., [1]Abass, O., [1]Longe, H. O. D.
[1]Computer Sciences Department, University of Lagos, Akoka-Lagos
[2]Computer Science Department, National Open University of Nigeria, Abuja
**Corresponding author: releafolorunso@gmail.com*

ABSTRACT
Distributed Denial of Service (DDoS) attack floods the network with loads of unwanted packets and requests that weigh down the system resources such as memory and processors. Hidden Markov model (HMM) is one of the models that can be used to predict and detect such attacks. A problem to be solved was determining the observable states and subsequently, the model parameters since the performance of the model depends on the accurate selection of these parameters. In this work, the concept of entropy was used to determine the observable states, which characterise the HMM. In order to improve computational efficiency of the algorithm for estimating the parameters of the model, Kullback-Liebler Divergence (KLD) method was employed for reducing and selecting appropriate observable states to achieve a good prediction model. The experimental results justified the suitability of KLD in reducing the entropy-based observable states of HMM for predicting DDoS attack.

Keywords: HMM, KLD, DDoS, Entropy, Observable States

1.0 INTRODUCTION
With the present astronomical growth of the Internet and the fact that most business transactions are performed online, the issue of security of network systems has become more prominent than before. This has led to more researches into techniques for protecting and safeguarding network systems. One of such areas of research is network attacks prediction. There are several types of network attacks and these can be classified into four main categories (Sharma *et al.,* 2015):

i) ***Denial of Service (DoS):*** where an attacker makes network resources too busy to serve legitimate requests. Examples are mail bomb, apache, syn flood etc.
ii) ***Probing (Probe):*** In probing attack, the attacker scans a network device so as to gather information about weaknesses or vulnerabilities that can be

exploited to compromise the target system. Examples are nmap, saint, mscan, *etc*.

iii) ***User to Root (U2R):*** in this category, an authorized user attempt to abuse the vulnerabilities of the system in order to gain privilege of root user they are not authorized for. Example are perl, Fd-format, xterm, etc.

iv) ***Remote to Local (R2L):*** here, a remote user sends packets to a machine over the internet to gain access as a local user to a local machine i.e. the weaknesses of the system is exploited by an external intruder to access the privileges of a local user. Examples include phf, xlock, guest *etc*.

The various categories of network attacks aim at undermining the CIA (Confidentiality, Integrity and Availability) properties of the network (Sodiya *et al*., 2004). The network attack that is the focus of this paper is DoS, which aims at attacking the availability property of the network such that the system is weighed down attending to illegitimate requests to the extent that it cannot attend to legitimate requests by legitimate users.

Several methods such as Time series, Machine Learning (Seng *et al*., 2010; Zhang *et al*., 2012; Satpute *et al.,* 2013), Markov Chain (Shin *et al*,. 2013), Hidden Markov Model (HMM) (Cheng *et al.,* 2012; Sendi *et al.,* 2012), Statistical Profiling (Saganowski *et al*., 2013), Data Mining (Sodiya *et al*., 2007), Neural Network (Saini *et al*., 2014), and combinations of these methods had been applied to detecting and predicting DDoS attacks (Siani *et al.,* 2014; Sharma *et al.,* 2015). Some of these methods have weaknesses such as false positives, low precision, high computational time, etc. For these reasons researches continue to evolve on how to improve on these weaknesses. However, among the aforementioned approaches, HMMs have been proved to be very promising for anomaly prediction over several other techniques because of their high accuracy in identifying attacks (Badajena *et al.,* 2012). However, the efficiency of HMM-based algorithms is hindered by long training time during the construction of the models (Sendi *et al*., 2012).

In this paper, an attempt is made to improve the performance of HMM by using the Kullback-Liebler Divergence (KLD) method to reduce the observable states of the model. The motivation for using the KLD-enhanced HMM approach is to improve the rate of convergence of the prediction model thereby reducing the training time as well as computational time as compared to using ordinary HMM approach. The quality of our proposed method was evaluated using DARPA datasets.

DDoS progresses in stages and can therefore be said to have different phases. At each phase there are some observable events that occur and these events can be used to predict the state of the system and what could happen in the system in the foreseeable future (Afolorunso *et al*., 2016). According to the experiments run by the MIT Lincoln Lab (MIT Lincoln Lab, 2000), DDoS attack session can be grouped into five phases as follows: (Lee *et al.*, 2008)

1) IPsweep to the DMZ (demilitarized zone) hosts from a remote site.
2) Probe of live IP's to look for the sadmind daemon running on Solaris hosts.
3) Breaks-in via the sadmind vulnerability, both successful and unsuccessful on those hosts.
4) Installation of the Trojan mstream DDoS software on three hosts in the DMZ.
5) Launching the DDoS.

Also, Lee *et al.,* (2008) established nine features that could be used in analyzing the characteristics of the network during a DDoS attack. The features are: *Entropy of source IP address, Entropy of source port number, Entropy of destination IP address, Entropy of destination port number, Entropy of packet type, Occurrence rate of Packet type (ICMP, UDP, TCP-SYN)* and *Number of packets*.

Afolorunso *et al.*, (2016) established that the aforementioned features can be used as observable states in formulating an HMM for predicting DDoS.

This paper proposes an HMM-based approach where these features are the observable states while the phases of the DDoS attack form the hidden states of the model. To further enhance the model and reduce the computation time, KLD was used to reduce the number of observable states of the model. The original model containing all the features as well as the KLD-enhanced HMM were trained and used to predict attacks. The performance of the two models were compared and the results reported.

The rest of the paper is organised as follows. Section 2 discusses related research that uses entropy and HMM for attack detection and prediction, Section 3 describes the research methodology, Experimental Results and Discussion are presented in Section 4. Section 5 presents the paper conclusion.

2.0 RELATED RESEARCH

For modeling a large number of temporal sequences, HMM can be an excellent tool, because it has been widely used for pattern matching in speech recognition

(Rabiner, 1989), image identification (Bunke, 2001), and network attacks (Cuppens, 2001). Warrender *et al.,* (1999) introduced HMM into anomaly detection for the first time. If an attack is considered to be a pattern of an observed sequence, HMM will be appropriate for mapping those patterns to one of many attack states. Several researchers have used HMM in one form or the other to either detect or predict network attack. Some of such works are discussed below:

Berezinski *et al.,* (2015) using data mining techniques proved that an entropy-based approach is suitable to detect modern botnet-like malware based on anomalous patterns in the network.

Saini *et al.,* (2014) gave a comprehensive review of works that deployed HMMs in network attacks detection and prediction.

Agarwal *et al.,* (2012) proposed a hybrid model that combines entropy-based IDS with Support Vector Machine-based system to detect network attack. DARPA dataset was used in evaluating the model. It was established that the hybrid model give fewer false alarm.

Sendi *et al.,* (2012) worked on an HMM architecture to predict intrusions and trigger good response strategies. A novel alert correlation was employed in decreasing false negatives in the prediction. Experimental results on the DARPA 2000 data set showed that the model can perfectly predict DDoS attacks and has a potential to detect multi-step attacks missed by the detection component.

3.0 RESEARCH METHODOLOGY

As mentioned earlier, in this research, we aim at improving the computational time of the original HMM algorithm. This is achieved by combining KLD and HMM algorithms to predict DDoS attacks. This section briefly presents the research model of this study and the proposed procedure for prediction.

Our research methodology consists of four major steps: in the first step, the network states are defined by means of clustering the traffic of the network. In the second step the initial probability distribution, the state transition probability and the emission transition probability of the HMM is built based on the definitions got from Step 1. In the third step the HMM is trained using the DARPA 2000 intrusion data set after which two sets of test data (DARPA 1999 no attack data set and simulated real time data set) are used to test the model and make predictions. In the fourth step KLD was used to reduce the observables state space

of the HMM using the algorithm in Fig. (4). This enhanced model was also trained and used for prediction. Finally the results and computational efficiency of the two models were compared. Fig. (1) is a pictorial overview of the proposed model.

3.1 Step 1: Defining the Network States

This step extracts the desirable features of the temporal network data using the Shannon Entropy (Berezinski *et al.*, 2015).

3.1.1 Shannon Entropy.

The definition of entropy as a measure of disorder comes from thermodynamic and was proposed in the early 1850s by Clausius *et al.*, (as cited in Berezinski *et al*., 2015*).* Shannon (1948) adopted entropy to information theory as a measure of the uncertainty associated with a random variable. The more random the variable, the bigger the entropy and vice versa. For a probability distribution $p(Y = y_i)$ of a discrete random variable Y, the Shannon entropy is defined as:

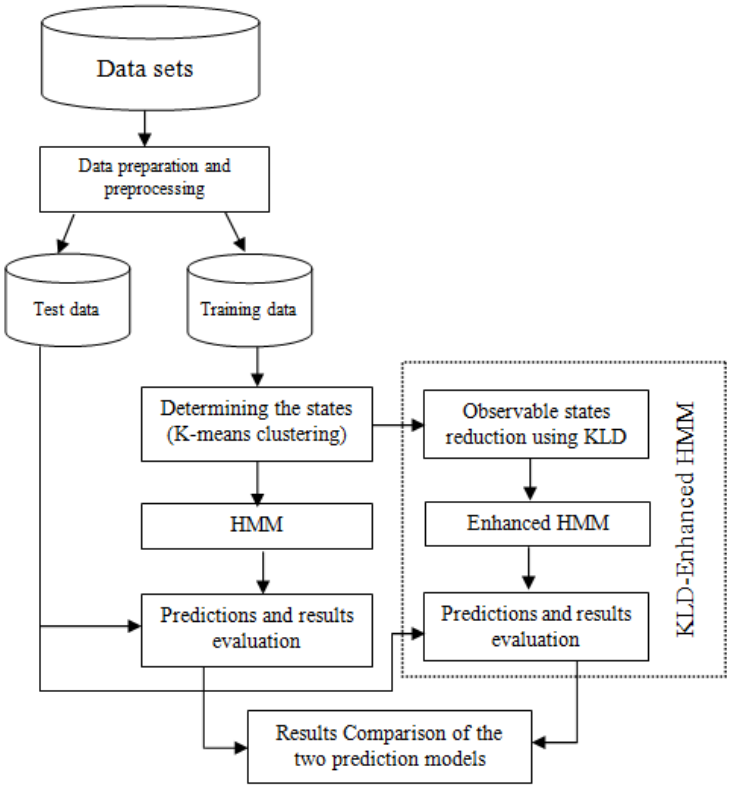

Fig. 1: Model architecture

$$H_s(Y) = \sum_{i=1}^{n} p(y_i)\log_a \frac{1}{p(yi)} \quad (1)$$

Y is the feature that can take values $\{y_1 \ldots y_n\}$ and $p(y_i)$ is the probability mass function of outcome y_i. The entropy of y can also be interpreted as the expected value of $\log_a \frac{1}{p(yi)}$

where Y is drown according to probability mass function p(y). Depending on the base of the logarithm, different units such as bits (a = 2), nats (a = e) or hurtleys (a = 10) can be used. For network attack detection/prediction, typically sampled probabilities estimated from a number of occurrences of y_i in a time window t are used. The value of entropy depends on randomness and the value of n. In order to measure randomness only, normalized forms, as used in this paper, have to be employed. For example, an entropy value can be divided by n or by maximum entropy defined as $\log_a(n)$. See Fig. (2) below for the algorithm for calculating normalized entropy.

After this, *K*-means clustering algorithm is applied to classify the network behaviour into states. Six states were generated for our work and as a result, the state of each observation can be represented by the cluster it belongs to.

Algorithm 1: Calculating normalized entropy

1) **Input**: Network Traffic
2) **Output**: Normalized entropy for each network feature
3) Loop: each interval of time until the traffic comes
   - 3.1) Extract features from packet header
   - 3.2) loop: for each packet in the time interval
     - 3.2.1) Calculate frequency of all distinct x // where x = source IP, destination IP, source port number, destination port number, packet type, packet size
     - End
   - 3.3) Loop: for each distinct x
     - 3.3.1) Calculate probability for each distinct x
       - $P_i = m_i/N$ // m_i = frequency of the i^{th} x; N = total number of packets in that time interval
     - 3.3.2) Calculate entropy for each distinct x
       - $h_i = -\sum_{i=1}^{n} P_i \log P_i$
     - End
   - 3.4) Normalize the entropy, in the time interval by
     - $H = -\sum_{i=1}^{n} h_i/\log(F)$ // F is total number of distinct x

END

Fig. 2: Normalized entropy calculation algorithm

3.2 Step 2: Estimating Model Parameters

Here, the HMM parameters were estimated from the training data set (DARPA 2000 data set). It should be noted that the model formulated here has six hidden states and six observable symbols.

3.2.1 Hidden Markov Model (HMM)

HMM, the simplest form of Dynamic Bayesian Network (DBN), is a doubly stochastic process: an unobservable (hidden) process *S*, which can only be observed through another (observable) stochastic process *O*. Each state in *Q* (the set *S* of hidden states) has state-transition probabilities (which are not visible) and a probability distribution over the possible values of *O*. The basic assumption in HMM is that the current hidden state of the system is affected only by its previous state.

An HMM is characterized by the following:

1) A finite set of *N* states (S = {s_1, s_2, …, s_N}). The states used in this paper corresponds to the phases of DDoS attack as listed in Section 1.0 above denoted by *I, P, B, T* and *D* respectively. A normal state, *N*, indicating that there is no malicious activity or any attempt to break into the system is added to the states. So, S_i = ($s_1 = N$; $s_2 = I$; $s_3 = P$; $s_4 = B$; $s_5 = T$; $s_6 = D$)

The relationship among these states is as diagrammatically shown in Fig. (3) below:

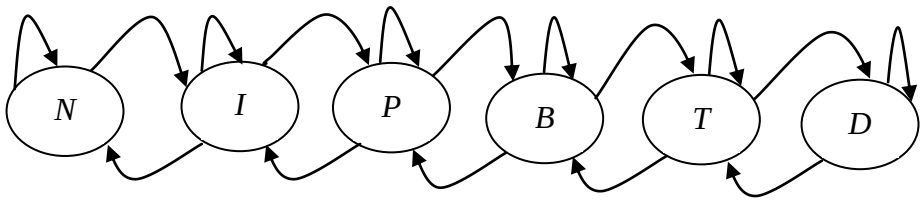

Fig. 3: Hidden Markov models states for prediction

2) A finite set of *M* possible symbols ($O = \{o_1, o_2, o_3, \ldots, o_M\}$). In our case, the first six of the features from the network traffic as listed in Section 1.0 above within certain time interval are the observations denoted by *IS, ID, PS, PD, PT, PO* respectively.

3) State Transition Probability (A_{ij}): A square matrix where a_{ij} is the probability that the system goes from state s_i to s_j

4) Emission Transition Probability ($B_j(k)$): A rectangular matrix where $b_j(o_k)$ is the probability that the symbol o_k is emitted when the system is in state s_j.

5) Initial State Probability (π_i): a row vector depicting the probability that the system starts in state s_i.

Since the states and output sequence are understood, it is customary to denote the parameters of an HMM by $\lambda = (A, B, \pi)$.

One of the aims of using an HMM is to deduce the likelihood of an attack of a specific type, given the set of observables contained in an example corresponding to an attack (Saini, 2014).

3.3 Step 3: Model Training and Testing

The model formulated in Step 2 is then trained until convergence. Then the two sets of test data as mentioned before were used to test the model and make predictions.

3.4 Step 4: Reducing the Observable States Space of the Model

3.4.1 Kullback-Liebler divergence method.

For discrete probability distributions P and Q, the KLD of Q from P is defined to be:

$$(P||Q) = \sum_i^n P(i) \ln \frac{P(i)}{Q(i)}$$ (Aczél **&** Daróezy, 1975) (2)

It can be described as the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P. The KLD is only defined if $Q(i) = 0 \Rightarrow P(i) = 0$, for all i (absolute continuity). If the quantity 0ln0 appears in the formula, it is interpreted as zero since $\lim_{x \to 0} x \ln(x) = 0$ (Aczél **&** Daróezy, 1975).

Algorithm 2: Implementing Kullback-Liebler Divergence for reducing Observable state space

Step 1: Load DARPA 1999 attack free training data and DARPA 2000 Intrusion Specific Datasets
Step 2: Load the results of Step 3.3.1 in Algorithm 1 above
Step 3: Compute: $DKL(P||Q) = \sum_i^n P(i) \ln \frac{P(i)}{Q(i)}$
Where:
For Case (a): P = Source IP address and Q = Destination IP address
For Case (b): P = Source port and Q = Destination port
For Case (c): P = Occurrence rate of packets per unit time and Q = Packet type

If the divergence is not significant, i.e. if it tends towards zero, then either P or Q can be used in each case

Else use both P and Q

Fig. 4: Implementation of KLD algorithm for HMM observable symbols

Here, the KLD was used to reduce the number of observable states of the HMM using the algorithm in Fig. (4). The newly obtained HMM was also trained and used for prediction as in Step 3 above.

The results obtained from the two models were then compared.

4.0 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the experimental results of the system is evaluated. The training and test data are available at http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html, https://www.ll.mit.edu/ideval/data/.

4.1 HMM Parameters

At first, that is, at system start-up, $\pi = (1, 0, 0, 0, 0, 0)$, which implies that the system is in the normal state with 100% probability. Next the state transition probability (A), which is a 6 X 6 matrix and the emission probability matrix (B), also a 6 X 6 matrix was estimated from the temporal network.

The HMM, $\lambda = (A, B, \pi)$, was trained using Baum-Welch algorithm (Ibe, 2013), the model converged after 90 iterations. Two sets of test data, as earlier mentioned, were run through the model for prediction using Viterbi algorithm (Ibe, 2013). Fig. (6) and Fig. (7) show the results of the convergence rate compared with that of the KLD-enhanced model.

4.2 KLD Enhanced-HMM Parameters

The π, initial probability distribution, and the state transition probability (A) remained the same as in the original HMM above. However, the probabilities of

the observable states space is reduced using the algorithm in Fig. (4) above. The results obtained showed that three of the observables namely: entropy of source IP (*SI*), entropy of destination IP (*DI*) and Occurrence rate of Protocol (*PO*) can be used to represent the system. Fig. (5) shows the relative entropy distribution of the reduction process. The resulting emission probability matrix (B) is a 6 X. 3 matrix.

The new model, like the original HMM, was likewise trained and used for prediction using the same sets of data. It was observed that the new model converged faster (after 60 iterations) than the previous one. See Table (1), Fig. (6) and Fig. (7).

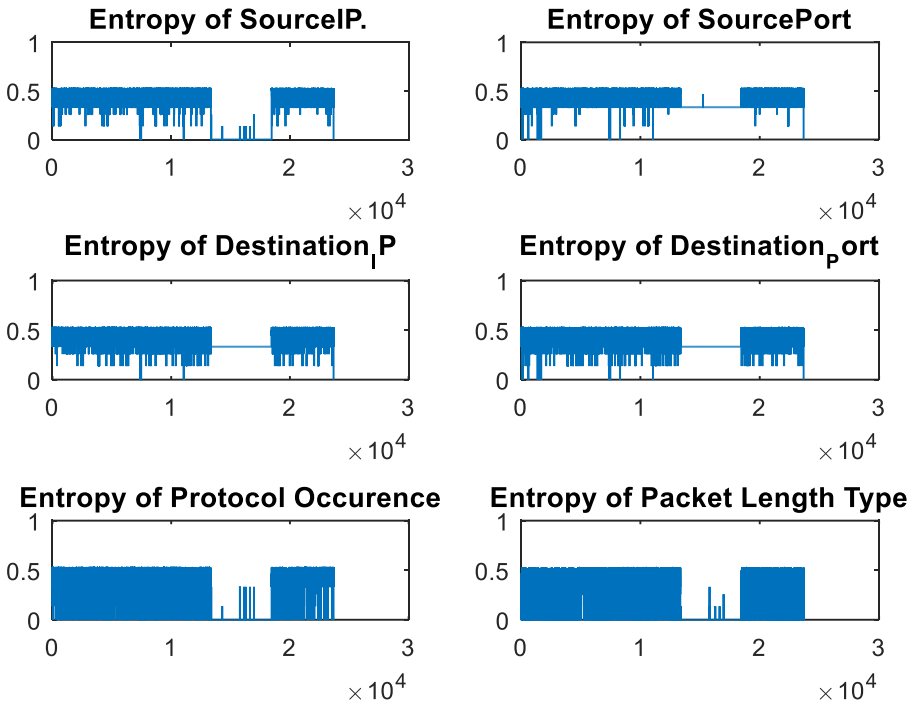

Fig. 5: Relative entropy distribution of the observable state space

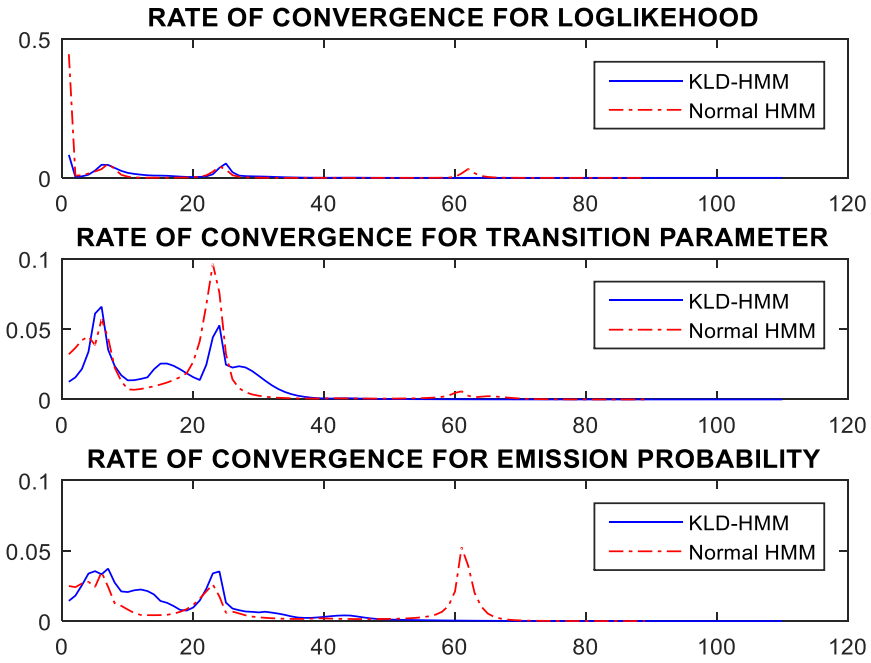

Fig. 6: Comparison of the rate of convergence of the two models

Table (1) shows the performance benchmark of the two models as defined by Kumar (2014) and Wu & Banzhaf (2010). It was observed that the KLD-HMM model converges faster, (after about 60 iterations in 59.53 seconds) than the HMM that converged after about 200 iterations in 714.90 seconds. The confusion matrices, which represent true and false classification results (Kumar, 2014) of the KLD-HMM and HMM models are given by $\begin{pmatrix} 0.84 & 0.16 \\ 0.21 & 0.79 \end{pmatrix}$, $\begin{pmatrix} 0.77 & 0.23 \\ 0.29 & 0.71 \end{pmatrix}$, respectively. This means that the KLD-HMM model has 84% true positive rate (TPR), 16% false negative rate (FNR), 79% true negative rate (TNR) and 21% false positive rate (FPR) with prediction accuracy at 82% while the HMM model has 77% true positive rate (TPR), 23% false negative rate (FNR), 71% true negative rate (TNR) and 29% false positive rate (FPR) with prediction accuracy at 74%. See Table (1).

Table 1: Performance benchmark of the models

MODELS	Computational time in seconds	True Positive Rate (TPR)	False Negative Rate (FNR)	False Positive Rate (FPR)	True Negative Rate (TNR)	Accuracy
KLD-HMM	59.53	0.84	0.16	0.21	0.79	0.82
HMM	714.90	0.77	0.23	0.29	0.71	0.74

Fig. (8), the Receiver operator characteristics (ROC) curve of the test data, is a graphical metric that illustrates the performance of a classifier which in our case is an HMM model that classifies Packet sequence as Threat or Normal traffic. The plot shows the rate of prediction as against false alarm rate. The curve with the continual variation depicts the plot of the HMM model and it shows an approximate variation between false and true classification of sequence packet data. The other curve representing the KLD-enhanced HMM model shows a less accurate detection rate initially until a threshold (around 0.005) is overcome where the performance of the model becomes excellent.

In the implementation of a DDoS attack Prediction System, this threshold value that translates to an improved performance should be taken into account when developing such systems.

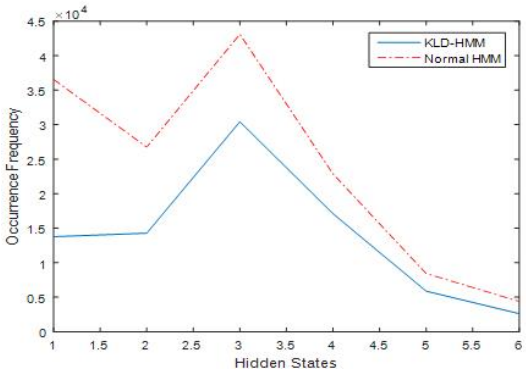

Fig. 7: Frequency distribution per state of the two models

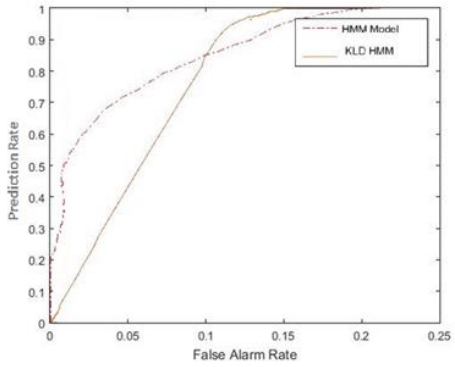

Fig. 8: ROC curve of the performance of the two models

5.0 CONCLUSION

In this paper, we presented an architecture to predict DDoS attack. Our experimental results on the DARPA 2000 data set have shown that our model converges faster, which means computational efficiency, and shows good

performance in predicting the attacks. In future we will further improve our model by optimising the training algorithm, Baum-Welch, using a technique that will make it converge to global maxima. We also plan, if possible, to interface our system with live data in real time.

REFERENCES

Aczél, J. & Daróezy, Z. (1975). *On measures of information and their characterizations*, New York-San Francisco-London. Academic Press. XII, 234 S., (Mathematics in Science and Engineering 115)

Afolorunso, A. A., Abass, O., Longe, H. O. D. & Adewole, A. P. (2016), Forecasting Distributed Denial Of Service Attack Using Hidden Markov Model, A paper accepted for publication in International Journal of Biological and Physical Sciences**.** *Published by Faculty of Pure and Applied Sciences, LAUTECH.* Website: ***www.sciencefocusngr.org***

Agarwal, B. & Mittal, N. (2012). Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques, *Procedia Technology*, *6*(2012), 996-1003.

Badajena, J. C. & Rout, C. (2012). Incorporating hidden Markov model into anomaly detection technique for network intrusion detection. *International Journal of Computer Applications*. *53*, 42-47.

Berezinski, P., Jasiul, B. & Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy 2015*, *17*, 2367-2408. doi:10.3390/e17042367

Bunke, H. & Caelli, T. (Eds.). (2001). *Hidden Markov models: Applications in computer vision.* World Scientific, Series in Machine Perception and Artificial Intelligence, *45*.

Cheng, X. & Yangdan, N. (2012). The research on dynamic risk assessment based on hidden Markov models. *International Conference on Computer Science & Service System (CSSS).* 1106-1109. doi:10.1109/CSSS.2012 .280

Cuppens, F. (2001). Managing alerts in a multi-intrusion detection environment. In ACSAC '01 *Proceedings of 17th Annual Computer Security Applications Conference.* Retrieved from https://www.acsac.org/2001/papers/70.pdf

Dorogovs, P., Borisov, A. & Romanovs, A. (2011). Building an intrusion detection system for it security based on data mining techniques. *Scientific Journal of Riga Technical University*, 49, 43-48.

Flores, J. J., Antolino, A. & Garcia, J. M. (2010). Evolving hidden Markov models for network anomaly detection. *Sixth International Conference on Networking and Services (ICNS)*. doi: 10.1109/ICNS.2010.44.

Haslum, K., Moe, M. E. G. & Knapskog S. J. (2008). Real-time intrusion prevention and security analysis of networks using HMMs. *33rd IEEE Conference on Local Computer Networks*, Montreal, Canada. **doi:** 10.1109/LCN.2008.4664305

Ibe, O. C. (2013). *Markov processes for stochastic modelling* (2nd ed.). Burlington, MA: Elsevier Academic Press.

Kumar, G. (2014). Evaluation metrics for intrusion detection systems - A study. International *Journal of Computer Science and Mobile Applications*, *2*(11), 11-17

Lee, K., Kim, J., Kwon, K. H., Han, Y. & Kim, S. (2008). DDoS attack detection method using cluster analysis. *Expert Systems with Applications. 34*, 1659-1665.

Liao, S. H., Chu, P. H. & Hsiao, P. Y. (2012). Data mining techniques and applications; A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*, 11303-11311.

MIT Lincoln Lab (1999). DARPA intrusion detection scenario specific datasets. Available at <http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_ index.html>.

MIT Lincoln Lab (2000). DARPA intrusion detection scenario specific datasets. Available at <http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_ index.html>.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Saganowski Ł., Goncerzewicz M. & Andrysiak T. (2013). Anomaly detection preprocessor for SNORT IDS system. In: Choraś R. (Eds.) *Image Processing and Communications Challenges 4*. Advances in Intelligent Systems and Computing, Springer, Berlin, Heidelberg. *184*, 225-232. doi: 10.1007/978-3-642-32384-3_28

Saini, P. & Godara, S. (2014). Modelling intrusion detection system using hidden Markov model: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(6), 542-547. (Available online at: www.ijarcsse.com)

Satpute, K., Agrawal, S., Agrawal, J., & Sharma S. (2013). A Survey on Anomaly Detection in Network Intrusion Detection System Using Particle Swarm Optimization Based Machine Learning Techniques. In:

Satapathy S., Udgata S., & Biswal B. (Eds.). *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*. Advances in Intelligent Systems and Computing. Springer, Berlin, Heidelberg. *199*, 441-452.

Sendi, S., Dagenais, M. Jabbarifar, M. & Couture, M. (2012). Real time intrusion prediction based on optimized alerts with hidden Markov model. *Journal of Networks*, *7*(2), 311-321.

Seng, J. L. & Chen, T. C. (2010). An analytic approach to select data mining for business decision. *Expert Systems with Applications*, *37*, 8042-8057.

Shannon, C. (1948). *A Mathematical Theory of Communication.* Bell Syst. Tech. J. *27*, 379–423.

Sharma, S. & Gupta, R. K. (2015). Intrusion detection system: A review. *International Journal of Security and Its Applications 9*(5), 69-76.

Shin, S., Lee, S., Kim, H. & Kim, S. (2013). Advanced probabilistic approach for network intrusion forecasting and detection. *Expert Systems with Applications*. *40*, 315-322.

Sodiya, A. S., Longe, H. O. D. & Akinwale, A. T. (2004). A new two-tiered strategy to intrusion detection. *Information Management and Computer security, 12(*1*),* 27 - 44.

Sodiya, A. S., Adeniran, O. & Ikuomola A. J. (2007). An expert system-based site security officer, *Journal of Computing and Information Technology - CIT 15*(3), 227–235.

Warrender, C., Forrest, S. & Pearlmutter, B. (1999). Detection of intrusion using system calls: Alternative data models[C]. *IEEE Symposium on Security and Privacy*. IEEE Computer Society

Wu S. X. & Banzh F. W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing Journal*, *10*(1), 1-35

Zhang, X., Jia, L., Shi, H., Tang, Z. & Wang, X. (2012). The application of machine learning methods to intrusion detection. *Congress on Engineering and Technology (S-CET)*, Spring, 1-4. doi**:** 10.1109/SCET.2012.6341943