OPTIMISATION OF HIDDEN MARKOV MODEL FOR DISTRIBUTED DENIAL OF SERVICE ATTACK PREDICTION USING VARIATI ONAL BAYESIAN

[1]Afolorunso, A. A., [2]Abass O.

[1]Faculty of Science, National Open University of Nigeria, Lagos, Nigeria
[2]Department of Computer Sciences, University of Lagos, Lagos, Nigeria

ABSTRACT
Distributed Denial of Service (DDoS), is a coordinated attack that is majorly carried out on a massive scale against the availability of services property of a target system or network resources. Due to the continuous evolution of new attacks and ever-increasing number of vulnerable hosts on the Internet, several DDoS attack detection, prevention or prediction techniques have been proposed. Some of these techniques have shortcomings such as high false positive rate, high computational time, low prediction precision and so on. In order to overcome these shortcomings, researches are being carried out to improve on the existing systems. This paper, which is one of such efforts to improve on the performance of existing DDoS attack prediction methods, presents a novel learning method based on Variational Bayesian (VB) algorithms to obtain an Hidden Markov Model (HMM) with optimized number of states in the HMMs and its model parameters for DDoS attack prediction. This method not only overcomes the shortcomings of the slow convergence speed of the HMM approach, but it also avoids the problem of overfitting the model structure by removing excess transition and emission processes. From the experiments with the DARPA 2000 intrusion specific datasets, this method is able to find the optimal topology in every case. The experiments show that the VB-HMM approach has a better average precision rate than the HMM trained by the Baum-Welch method. This shows that the VB-HMM method is better optimized than the HMM trained by the Baum-Welch method.

Keywords: *DDoS, Variational Bayesian, Hidden Markov model, network attacks*

1.0 INTRODUCTION

With the increase in global interconnectivity via the Internet comes the challenge of security/protection of connected systems. Vulnerability of inter-networked systems has been exerbated by new paradigms such as Internet of Things (IoT) and Internet of everything (IoE). In order to respond to the challenges, researches into techniques for protecting and safeguarding network systems continue to emerge. One of such research areas is network attacks prediction. The different types of network attacks 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 include mail bomb, apache, syn flood
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 include nmap, saint, mscan.
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 he/she is not authorized for. Example include perl, Fd-format, xterm.
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.

The various categories of network attacks aim at undermining the CIA (Confidentiality, Integrity and Availability) properties of the network (Sodiya, *et al.*,2004). But specifically, Distributed Denial of Service (DDoS), which is a type of **DOS attack** where multiple compromised systems, which are often infected with a Trojan, are used to target a single system leading to unavailability of the system services/resources to legitimate users. Commercial web servers, banks, educational institutions and government websites are usually major victims of such attacks. A typical instance and more recent occurrence of DDoS attack is the large-scale DDoS against New Hampshire-based Internet performance company, Dyn, which caused major Internet disruptions on Friday, 21st October, 2016. The attack disrupted internet service across Europe and United States of America (USA). Users were unable to access many major websites such as

Twitter, Spotify, Netflix, Amazon, Tumblr, Reddit and other sites (**USATODAY**, *EDT October 21, 2016)*

Several of the techniques 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 (Jiao, 2012), Neural Network, and combinations of these methods, which had been applied to detecting and predicting DDoS attacks (Siani, *et al,* 2014; Sharma, *et al*. 2015) have weaknesses which include false positives, low prediction precision, high computational time and false negatives. According to For these reasons, efforts continue to evolve on how to improve on these weaknesses. However, among the aforementioned approaches, HMMs, which is a kind of hybrid techniques that incorporate time series and probabilistic techniques, 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, research has shown that the efficiency of HMM-based algorithms is hindered by long training time during model construction (Sendi, *et al,* 2012). This study aims at overcoming this limitation by employing Variational Bayesian Inference (VB) in optimizing the HMM learning algorithm.

According to Lee, *et al (*2008), DDoS progresses in stages and can therefore be said to have different phases. the experiments run by the MIT Lincoln Lab (2000) partitioned DDoS attack session into five phases as follows:

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.

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).

Lee, *et al* (2008) also identified nine features viz. *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* that could be used in analyzing the characteristics of the network during a DDoS attack.

A DDoS attack prediction system is expected to predict the possibility of attack in time for steps to be taken to avert it without adding too much overhead in terms of resources consumption, which might adversely affect the performance of the system.

In this study, the Variational Bayesian (VB) inference algorithm is employed to develop a novel parsimonious and computationally efficient model for predicting DDoS attacks in network systems. The rest of this paper is organised as follows: Section two presents previous relevant works to this study; Section three gives the proposed research methodology; Section four presents the experimental results and discussions of the proposed model; while Section five presents the conclusion of the study and future work.

2.0 RELATED RESEARCH

HMM, which is an excellent tool when it comes to modeling large number of temporal sequences, has been widely used for pattern matching in speech recognition (Rabiner, 1989), image identification (Bunke, 2001), diagnotics (Nkemnole, et al, 2013) and network attacks (Cuppens, 2001). Since its introduction into anomaly detection by Warrender, *et al.* (1999), HMM has been deployed either singly or in combination with other techniques in network anomaly detection and prediction. Some of such works are discussed below:

Haslum, et al (2009) used an HMM model that models only integrity and confidentiality, and make no attempts to model availability. They believe that availability is best modeled separately. Preliminary experimental results from this system indicates that the proposed framework is efficient for real-time distributed intrusion monitoring and prevention.

Khosronejad, et al (2013) worked on a hybrid approach for modeling IDS. C5.0 and HMM were combined as a hierarchical hybrid intelligent system model. Experimental results with KDD Cup 99 benchmark Intrusion data showed that the proposed hybrid system provide more accurate intrusion detection compared to ordinary HMM approach

Rao, et al (2012) applied HMM to monitor Application Layer DDOS attacks on web servers.

They applied forward-backward algorithm to train HMM model thereby increasing the response time of the application. In their work, which is a counter-solution to diverse Application layer DDOS attacks, the web site design was customised so as to minimise Application layer DDOS attacks.

Divya, et al. (2015) proposed an hybrid framework, which combined two machine-learning techniques, hidden Markov model (HMM) and genetic algorithm (GA) for predicting future intrusion attacks in network systems. As indicated, the framework was made up of two main components: the first component uses GA to formulate efficient intrusion detection rules which leads to a precise attacks detection, the second component employs HMM in predicting the next attack plan of the attacker. The combination of these two gives a good intrusion prediction capability with reduced false positive rate.

Udaya et al. (2016) proposed an HMM-based alert prediction framework. Alert clustering was employed to group selected alert attributes together. A given sequence of alerts is converted to a sequence of alert clusters and then HMM is used to predict future alert clusters based on the input. The proposed technique also provided the alert category as well as the source IP address, the destination IP address, and the alert type, which are critical in responding to an intrusion. From the experimental results, it was observed that a smaller number of clusters improves prediction accuracy. A small number of clusters resulted in a smaller set of unique symbols for the HMM model which improves the learning abilities of the HMM model compared to a larger symbol size. However, smaller number of clusters hinders the separation of unique alert types and cause merging of two or more alert types. The experimental results also indicated that when the number of hidden states are lower than the number of observations, level 1 prediction accuracy is lower compared to higher number of hidden states. It was observed that when the number of hidden states are low, it may not be possible to model the system states changes efficiently because not enough states are available to represent state transition during a multi-stage intrusion scenario. However, they identified the following challenges to be addressed in the proposed alert prediction framework: (1) increasing the prediction accuracy with the increase of cluster size and predicting intrusion types that are not present in the training data set, and (2) identifying false alerts and misleading intrusion actions generated by the attacker in order to mislead intrusion detection systems.

3.0 RESEARCH METHODOLOGY

As earlier mentioned, this study aims at developing a novel parsimonious DDoS attack prediction model with high prediction precision and improved computational time. This is achieved by combining VB with HMM algorithms to predict DDoS attacks. This section briefly presents the research model of this study and the proposed procedure for prediction, which to the best of our knowledge no other work has used the combination of all the methods here in the same context.

The proposed model is based on HMM algorithms. The entropy-based features to be used as the observable states of the HMM are many so Kullback-Liebler Divergence (KLD) is used to select the minimum number of the features that could represent the whole to achieve improved performance without loss of information. Due to the shortcomings of HMMs especially the traditional learning algorithm of the HMM, VB is deployed in training the model.

The experimental procedure consists of four major steps. In the first step, the network states are defined by means of clustering the network traffic based on the entropy values of the network traffic features and the observables states of the model reduced using adapted relative entropy algorithm. In the second step, the parameters of the model, that is, the initial probability distribution, the state transition probability and the emission transition probability of the HMM is built based on the definitions got from the first step. In the third step, the traditional HMM algorithm is used to train the model formulated in step 2 using the DARPA 2000 intrusion dataset after which two sets of test data (DARPA 1999 (no attack) dataset and simulated real time dataset) are used to test the model and make predictions. In the fourth step, VB algorithm was used to train the HMM model of step 2. The VB-HMM was also tested and used for prediction. Finally, the results and computational efficiency of the two models were compared.

The architecture of the proposed model is as in Figure 1 below:

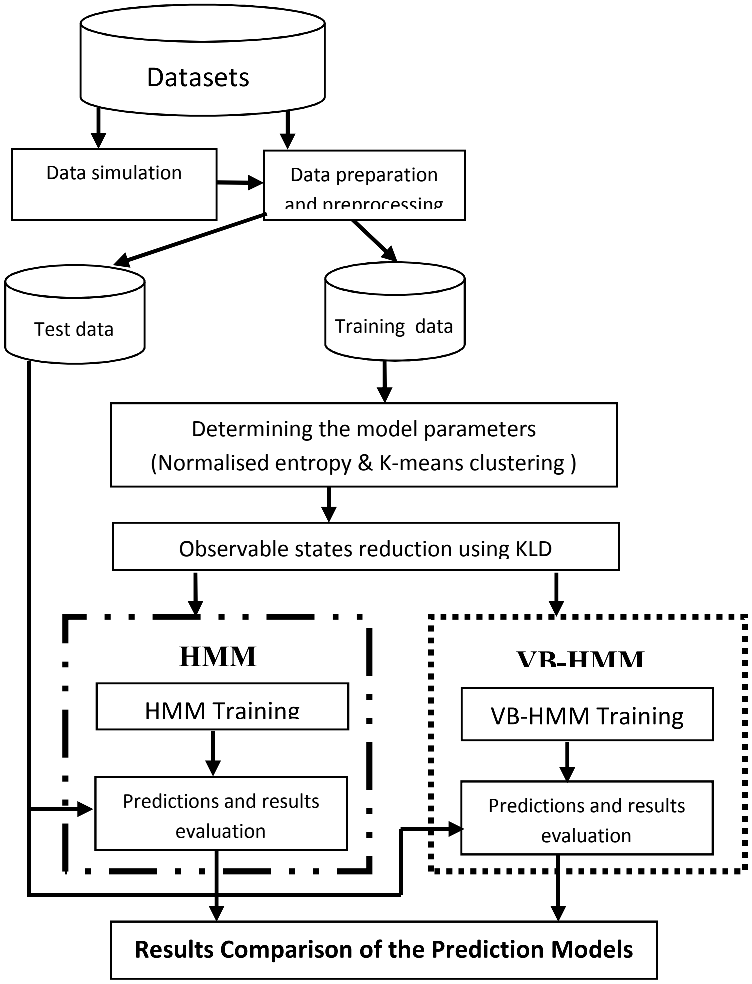
Datasets
Data simulation
Data preparation and preprocessing
Test data
Training data
Determining the model parameters (Normalised entropy & K-means clustering)
Observable states reduction using KLD
HMM
HMM Training
Predictions and results evaluation
VB-HMM
VB-HMM Training
Predictions and results evaluation
Results Comparison of the Prediction Models

Figure 1: The Proposed Experimental Framework

3.1 Model construction

3.1.1 Estimating the values of network features

For an information source with *n* independent symbols each with probability of choice *P*(*i*), the entropy, *H,* is defined as below (Shannon & Weaver, 1963):

$$H = -\sum_{i=1}^{n} P(i) \log_2 P(i) \qquad (1)$$

Therefore, entropy can be computed on a sample of consecutive packets. Comparing the value for entropy of some sample of packet header fields to that of other samples of packet header fields provides a mechanism for detecting and predicting changes in the randomness.

In order to construct the HMM, using the concept of entropy (Berezinski, *et al*, 2015), the desirable features of the temporal network data as listed in section 1 were estimated using the normalised entropy algorithm in Afolorunso, et al. (2016) at regular interval. To compute the entropies, the probabilities of each quantity in the training data was computed and plugged into equation (1)

Then, *K*-means clustering algorithm (MacQueen, 1967) is applied to classify the network behaviour into states. The state of each observation is identified by the cluster it belongs to. To achieve model parsimony, the adapted KLD algorithm in Afolorunso, et al, (2016) was then applied in reducing the observable states of the model.

3.1.2 Determining model parameters

The values of the estimated features in Section 3.1.1 above were then used in determining the HMM parameter λ = (A, B, π).

HMM is a type of finite state machine with a set of hidden states, $\boldsymbol{Q}$, an output alphabet (observations), $\boldsymbol{O}$, transition probabilities, **A**, output (emission) probabilities, $\boldsymbol{B}$, and initial state probabilities, $\boldsymbol{\Pi}$. The current model state is usually hidden and not observable but each state produces an output with a specific probability ($\boldsymbol{B}$). Usually the states, $\boldsymbol{Q}$, and outputs, $\boldsymbol{O}$, are understood, hence an HMM is customarily a triple, ($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}$). Traditionally, an HMM is characterized by the following:

i) Hidden states $\boldsymbol{Q} = \{ q_i \}, i = 1, . . . , N$.
ii) Transition probabilities $\boldsymbol{A} = \{a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)\}$, where $t = 1, . . . , T$ is time, and q_i in $\boldsymbol{Q}$. That is, $\boldsymbol{A}$ is the probability that the next state is q_j given that the current state is q_i.
iii) Observations (symbols) $\boldsymbol{O} = \{ o_k \}, k = 1, . . . , M$.
iv) Emission probabilities $\boldsymbol{B} = \{ b_{ik} = b_i(o_k) = P(o_k \mid q_i) \}$, where o_k in $\boldsymbol{O}$.
v) Initial state probabilities $\boldsymbol{\Pi} = \{p_i = P(q_i \text{ at } t = 1)\}$.

As in Afolorunso (2016), the five phases of DDoS resulting from Section 3.1.1 and an additional state *N* that represents the normal state when no malicious activity is going on in the system, forms the set of hidden states from which the parameters *A* and π are derived. These states are represented by the symbols, *I*, *P*, *R*, *T*, *D* and *N* respectively. Hence, $Q_i = (q_1 = N; q_2 = I; q_3 = P; q_4 = R; q_5 = T; q_6 = D)$.

3.1.3 Model training and testing

In training and testing the model, first, the model so formulated was trained using the Baum-Welch algorithm (Ibe, 2009) until convergence. Then the two sets of test data as aforementioned were used to test the model and make predictions. The prediction module is implemented using Viterbi algorithm (Ibe, 2009). Secondly, in pursuance of good performance in overall computational time and prediction precision, the model derived in Section 3.1.2 was again trained using VB algorithms (Beal, 2003).

3.1.3.1 Variational Bayesian inference (VB)

In machine learning, VB is mostly used to infer the conditional distribution (also known as the posterior distribution) over the latent variables given the observations (and parameters). VB for HMMs seeks to minimise the divergence between the true posterior and an approximation in which the parameters and hidden variables are assumed independent, which assumption allows for a very efficient iterative solution (Beal, 2003; MacKay, 1997). The paramount idea is to pick a family of distributions over the latent variable with its own variational parameters ($q(Y_{1:m}|\text{V})$ and then find the setting of the parameters that makes q close to the posterior of interest. q is used with the fitted parameters as a proxy for the posterior. The closeness of the two distributions is measured with KLD as in Beal (2003).

The concept of KLD embedded in the VB used in optimizing the HMM learning algorithm is as given below:

$$\text{KL}(\text{q}||\text{p}) = \text{E}_\text{q}\left[\log \frac{q(Y)}{p(Y|x)}\right] \qquad (2)$$

Where $x = x_1, x_2, ..., x_\text{n}$ are the observations and $y = y_1, y_2, ..., y_\text{m}$ are the hidden variables

It is not easy to minimize the KLD as a function of variational distribution. But this can be achieved by maximizing the evidence lower bound (ELBO) of the function. ELBO is obtained by applying Jensen inequality (f (E[X] ≥ E[$f(X)$] when f is concave) on the log probability of the observations, (Beal, 2003)

$$\log p(x) = \log \int_y \; p(x, y) \qquad (3)$$

$$= \int_y \; p(x, y) \frac{q(y)}{q(y)}$$

(4)

$$= \log \left(\text{E}_\text{q} \left[\frac{p(x,Y)}{q(y)}\right]\right)$$

(5)

$$\geq \text{E}_\text{q} \left[\log p(x,Y)\right] - \text{E}_\text{q} \left[\log q(Y)\right]$$

(6)

Equation (6) is the ELBO and it is the same bound used in deriving the EM algorithm (Ibe, 2009). A family of variational distributions is chosen to make the expectations computable. In this study, Dirichlet distribution is chosen because it is the conjugate to the complete-data likelihood terms of the HMM (Beal, 2003). It can be shown that the difference between the ELBO and KLD is the log normalizer, which is what the ELBO bounds (Beal, 2003). Hence, minimizing KLD is the same as maximizing the ELBO.

Finally, the results obtained from each of the two models above were compared using standard metrics for intrusion prediction such as false positive (FP) rate, false negative (FN) rate, true positive (TP) rate, true negative (TN) rate, precision rate, confusion matrix.

4.0 RESULTS AND DISCUSSIONS

4.1 Implementation platform

In this section, the model architecture and the design methodology steps enumerated in Section 3 were implemented in suitable programming platform and the experimental results evaluated using appropriate metrics. The training data and one of the 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/.

4.1.1 The prediction models

Implementing the step 1 of the design methodology, the desirable network traffic features is calculated at regular interval. First, at regular interval of one second and then five seconds. It was discovered that the two intervals produced identical results. So, five seconds interval was stuck to. Six hidden states corresponding to the number of clusters were arrived at. The states correspond to the phases of DDoS attack as listed in Section 1 (denoted by *I, P, R, T* and *D* respectively) and an additional normal state (denoted by *N*) when there are no traces of malicious activity or any attempt to break into the system. So, $Q_i = (q_1 = N; q_2 = I; q_3 = P; q_4 = R; q_5 = T; q_6 = D)$.

The finite set of *M* possible symbols ($O = \{o_1, o_2, o_3, \ldots, o_M\}$) in this study, are the three (entropy of source IP (*SI*), entropy of destination IP (*DI*) and Occurrence rate of Protocol (*PO*)) that the KLD results shows are adequate in representing the system out of the nine network features listed in Section 1,

The State Transition Probability (A_{ij}), the Emission Transition Probability ($B_j(k)$) and the Initial State Probability (π_i) were obtained from the data and approximated to five decimal places. At system start-up, π = (0.97183, 0.02452, 0.00118, 0.00098, 0.00108, 0.00041), which implies that the system, has the probability of 0.97183 of being in state *N*; 0.02452 of being in state *I*; 0.00118 of being in *P*; 0.00098 of being in *R*; 0.00108 of being in *T*, and 0.00041 of being in state *D*. Next, the state transition probability (*A*), which is a 6 X 6 matrix and the emission probability matrix (*B*), also a 6 X 3 matrix was estimated from the temporal network as depicted below:

$$A = \begin{matrix} 0.95880 & 0.00002 & 0.03819 & 0.00002 & 0.00216 & 0.00082 \\ 0.03026 & 0.00001 & 0.96659 & 0.00311 & 0.00001 & 0.00001 \\ 0.00001 & 0.95527 & 0.04468 & 0.00001 & 0.00001 & 0.00001 \\ 0.00166 & 0.00404 & 0.00041 & 0.84611 & 0.14777 & 0.00002 \\ 0.00011 & 0.00011 & 0.00011 & 0.96471 & 0.00011 & 0.03484 \\ 0.00046 & 0.00001 & 0.00001 & 0.00151 & 0.00060 & 0.99741 \end{matrix}$$

$$B = \begin{matrix} 0.89179 & 0.10011 & 0.00810 \\ 0.58648 & 0.31794 & 0.09559 \\ 0.55745 & 0.33734 & 0.10521 \\ 0.00960 & 0.98828 & 0.00212 \\ 0.00011 & 0.99977 & 0.00011 \\ 0.50798 & 0.30650 & 0.18551 \end{matrix}$$

$$\pi = \begin{matrix} 0.97183 & 0.02452 & 0.00118 & 0.00098 & 0.00108 & 0.00041 \end{matrix}$$

The HMM, λ = (*A*, *B*, π), was trained as earlier stated in Section 3.1, the model converged after about 60 iterations in 59.52 seconds as shown in Table 1 below.

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)
VB-HMM	17.79	0.91	0.09	0.11	0.89
HMM	59.53	0.84	0.16	0.21	0.79

Two sets of test data, as earlier mentioned, were run through the model for prediction using the Baum-Welch algorithm (Ibe, 2009). It was discovered that the model has FNR of 16% and FPR of 21%.

The HMM was then trained with VB and used for prediction using the same sets of data. First the Maximum Likelihood (ML) algorithm was run to convergence, and then the VB algorithm was run from that point in parameter space to convergence. This was achieved by initialising each parameter's variational posterior distribution to be Dirichlet with the ML parameter as the mean and by arbitrarily setting strength to 6. For the VB algorithm, the prior over each parameter was a symmetric Dirichlet distribution of strength 4.

Note that as depicted in Table 1, where it takes HMM about 60 iterations to converge to a local optimum, it takes only about 20 iterations for the VB optimisation to converge to global optimum. This is expected since the VB is initialised to the ML parameters, and so has less work to do.

As shown in Table 1, the computation time was within 18 seconds; the false positive rate was considerably reduce to 8% and the false negative rate to 11%. Compared to the traditional HMM constructed in this study, VB-HMM shows better performance on all metrics used. The combination of the TPR, FPR, TNR and FNR form the confusion matrix for each of the models. For example the confusion matrices for VB-HMM, HMM are given as, $\begin{pmatrix} 0.92 & 0.08 \\ 0.11 & 0.89 \end{pmatrix}$, $\begin{pmatrix} 0.84 & 0.16 \\ 0.21 & 0.79 \end{pmatrix}$, respectively.

Table 1 depicts the performance benchmark of the models while Figures 2, 3 and 4 are the pictorial representation of their comparison based on confusion matrices, computational time and prediction accuracy, respectively.

As shown in Table 1 and Figures 2, 3 and 4 VB-HMM was better than HMM in terms of computational time, confusion matrix and prediction accuracy, respectively.

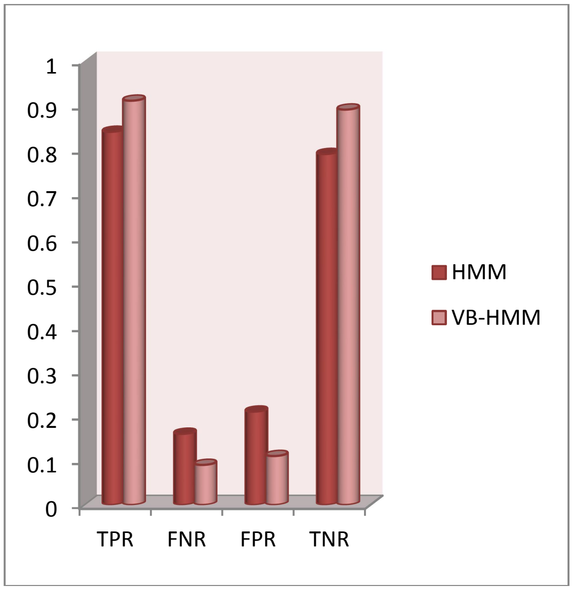

Figure 2: Graphical representation of the confusion matrices of the Models

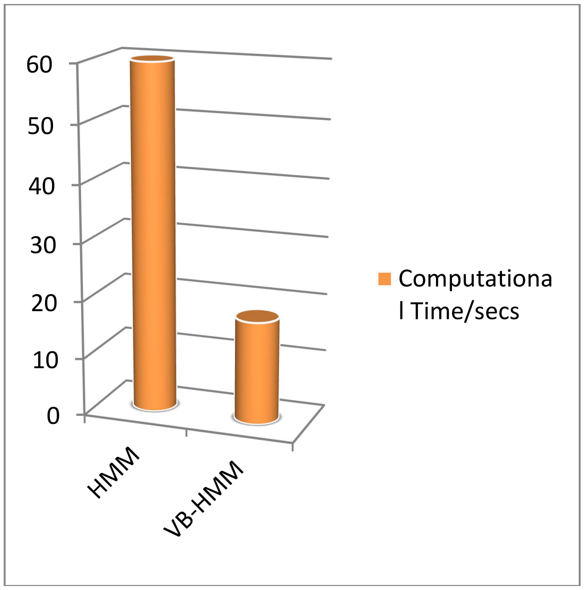

Figure 3: Graphical representation of the computational time of the Models

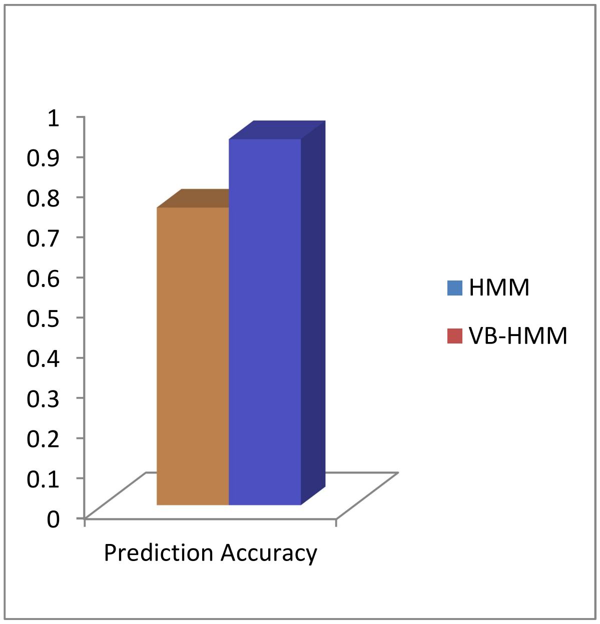

Figure 4: Graphical representation of the prediction accuracy of the models

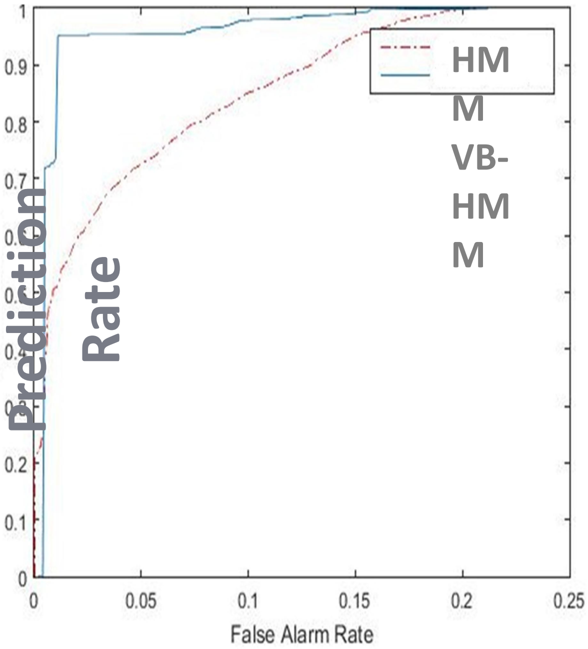

Figure 5: ROC curve of the performance of the HMM and VB-HMM

Figure 5, 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 VB-HMM model shows a less accurate detection rate initially until a threshold (around 0.01) 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.

This information depicts the trade-off between the models but it can be concluded that the VB-HMM model is more robust in terms of the prediction accuracy as depicted by its confusion matrix.

The VB-HMM model performed better in terms of classification of packet sequence as either normal or attack prone. Computationally, VB-HMM is more efficient as it reduces the propensity for over-fitting data due to model complexity which is not addressed by HMM.

Overall for real time application, the VB-HMM is recommended for use since it can compute and predict traffic status in a relatively shorter time. It also ensures the efficiency of prediction over all other models.

5.0 CONCLUSION

This study proposes a robust and efficient architecture for DDoS attack prediction. The proposed model was formulated, implemented, and tested with different types of datasets. Experimental results on the DARPA datasets have shown that the proposed model converges faster, which translates into computational efficiency, and shows good performance in predicting attacks compared to traditional HMM. In future, it is our plan to extend this work by using the proposed model to predict other types of network intrusions.

References:

Afolorunso, A.A., Adewole, A. P., Abass, O. , Longe, H. O. D. (2016). Kullback-Liebler Divergence for reducing the observable states space of Hidden Markov Model for predicting Distributed Denial of Service attack. 11th Unilag Conference and Fair.

Agarwal, B., Mittal, N. (2012), Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques, 2nd International Conference on Communication, Computing & Security [ICCCS-], Vol. 6

Badajena, J. C. and Rout, C. (2012), "Incorporating Hidden Markov Model into Anomaly Detection Technique for Network Intrusion Detection", International Journal of Computer Applications, vol. 53, pp. 42-47.

Beal, M. (2003). Variational Algorithms for Approximate Bayesian Inference. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London.

Berezinski, P. Jasiul, B. and Szpyrka, M. (2015). An Entropy-Based Network Anomaly Detection Method, Entropy 2015, 17 (An article)

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

Cheng, X. and N, Y.i (2012), "The Research on Dynamic Risk Assessment Based on Hidden Markov Models", Computer Science & Service System (CSSS), 2012 International Conference on, pp. 1106-1109.

Clausius, R. and Hirst, T. (1867). *The Mechanical Theory of Heat: With its applications to the steam-engine and to the physical properties of bodies*; J. van Voorst: London, UK

Cuppens, F. (2001), "Managing alerts in a multi- intrusion detection environment," in *Proc. 17th Annual Conf. Computer Security.*

Divya T., Muniasamy, K. (2015) Real-Time Intrusion Prediction Using Hidden Markov Model with Genetic Algorithm. In: Suresh L., Dash S., Panigrahi B. (eds) Artificial Intelligence and Evolutionary Algorithms in Engineering Systems. Advances in Intelligent Systems and Computing, Vol. 324. Springer, New Delhi

Dorogovs, P. Borisov, A. and Romanovs A. (2011), "Building an Intrusion Detection System for IT Security Based on Data Mining Techniques", Scientific Journal of Riga Technical University, vol. 49, pp. 43-48.

Flores, J. J., Antolino, A. and Garcia, J. M. (2010). Evolving Hidden Markov Models For Network Anomaly Detection. 10.1109/ICNS.2010.44. Sixth International Conference on Networking and Services (ICNS).

Haslum, K. Moe, M. E. G. and Knapskog S. J. (2008), "Realtime intrusion prevention and security analysis of networks using HMMs," 33rd IEEE Conference on Local Computer Networks, Montreal, Canada.

Ibe, O. C. (2009) Markov Processes for Stochastic Modelling, Elsevier Academic Press.

Jemili, F., Zaghdoud, M. and Ahmed, M. B. (2009*)* Hybrid Intrusion Detection and Prediction multiAgent System, HIDPAS, *(IJCSIS) International Journal of Computer Science and Information Security,Vol. 5, No.1,*

Kim, S., Shin, S., Kim, H., Kwon, K. and Han, Y. (2010), Hybrid Intrusion Forecasting Framework for Early Warning System, IEICE TRANS. INF. & SYST., VOL.E91–D, NO.5

Kumar, P. and Ravi, V. (2007). Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques, European Journal of Operational Research, 180 (1)

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

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

Lin, S. C. and Tseng S. S. (2004). Constructing detection knowledge for DDoS intrusion tolerance. Expert Systems with Applications, 27

MacKay, D. J. C. (1997). Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge.

MacQueen, J. B. (1967): "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability"*, Berkeley, University of California Press, 1:281-297

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

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>.

Nkemnole, E. B. Abass, O. and Kasumu, R. K. (2013): Parameter Estimation of a Class of Hidden Markov Model with Diagnostics". Journal of Modern Applied Statistical Methods 12(1): 181 - 197.

Pontes, E. (2012), "Distributed Multiagent Intrusion Forecasting System (DMIFS) Based Prediction Models for Cyber Attacks Detection System," In *Proceedings of the 7th European Symposium on Research in Computer Security*, Zurich, Switzerland, pp. 264-280

Saganowsk, M. G. i and Andrysiak, T. (2013), "Anomaly Detection Preprocessor for SNORT IDS System", Image Processing and Communications Challenges 4: Springer, (2013), pp. 225-232.

Saini, P. and Godara, S. (2014), "Modelling Intrusion Detection System using Hidden Markov Model: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6. (Available online at: www.ijarcsse.com)

Satpute, K., Agrawal, S., Agrawal, J. and Sharma, S. (2013), "A Survey on Anomaly Detection in Network Intrusion Detection System Using Particle Swarm Optimization Based Machine Learning Techniques", Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 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, Vol. 7, No. 2.

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

Sharma, S. and Gupta, R. K. (2015). "Intrusion Detection System: A Review", International Journal of Security and Its Applications Vol. 9, No. 5 pp. 69-76

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

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

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

Udaya S. K., Thanthrige, P. M., Samarabandu, J. and Wang, X. (2016). Intrusion Alert Prediction Using a Hidden Markov Model. https://arxiv.org/pdf/1610.07276

USATODAY, *EDT October 21, 2016*. www.usatoday.com/story/tech/2016/10/21/cyber-attack-takes-down.../92507806/

Warrender, C. Forrest, S. and Pearlmutter, B. (1999). Detection of Intrusion Using System

Calls: Alternative Data Models[C]. IEEE Symposium on Security and Privacy. IEEE Computer Society

Zhang, X., Jia, L., Shi, H., Tang, Z. and Wang, X. (2012), "The Application of Machine Learning Methods to Intrusion Detection", Engineering and Technology (S-CET), 2012 Spring Congress on, pp. 1-4.