Contemporary Sequential Network Attacks Prediction using Hidden Markov Model

Timothy Chadza[†,°], Konstantinos G. Kyriakopoulos[†,‡], and Sangarapillai Lambotharan[†]

[†]Wolfson School of Mech., Elect. and Manu. Engineering, Loughborough University, LE11 3TU, UK
[°]Department of Electrical Engineering, University of Malawi-Polytechnic, P/Bag 303, Blantyre, MW
[‡]Institute for Digital Technologies, Loughborough University London, London, E15 2GZ, UK
e-mail: tchadza@poly.ac.mw, {elkk, s.lambotharan}@lboro.ac.uk

***Abstract*— Intrusion prediction is a key task for forecasting network intrusions. Intrusion detection systems have been primarily deployed as a first line of defence in a network, however; they often suffer from practical testing and evaluation due to unavailability of rich datasets. This paper evaluates the detection accuracy of determining all states (AS), the current state (CS), and the prediction of next state (NS) of an observation sequence, using the two conventional Hidden Markov Model (HMM) training algorithms, namely, Baum Welch (BW) and Viterbi Training (VT). Both BW and VT were initialised using uniform, random and count-based parameters and the experiment evaluation was conducted on the CSE-CIC-IDS2018 dataset. Results show that the BW and VT count-based initialisation techniques perform better than uniform and random initialisation when detecting AS and CS. In contrast, for NS prediction, uniform and random initialisation techniques perform better than BW and VT count-based approaches.**

***Keywords*—Intrusion prediction, Hidden Markov Model, Baum Welch, Viterbi training, CSE-CIC-IDS2018 dataset**

I. INTRODUCTION

Cyber-attacks through intrusions have become widespread that detection of such intrusions is paramount. Currently, an Intrusion Detection System (IDS) remains a mandatory line of defence and is crucial in protecting critical networks from intrusions [1]. IDSs can either be anomaly-based or signature-based. The former leverages a model of the system under normal behaviour and, thereby, detects any potential deviations from normality, whereas the latter uses a database with signatures of known attacks to detect malicious activities [2]. Commercially, signature-based IDSs are widely used, but on the other hand, anomaly-based IDSs have the potential to detect unknown attacks. However, in general, anomaly-based IDSs have low detection and high false positive rates [3]. To enhance IDSs in detecting novel attacks, efficient adaptive Machine Learning (ML) algorithms are often used.

Hidden Markov Model (HMM), is a ML technique widely used for detection and prediction of cyber-attacks. Essentially, HMMs have been leading in both intrusion detection and multi-stage attack prediction [4]. Among other merits, HMM can detect unknown intrusions, predict potential future steps of an the intruder, and process data streams on-the-fly in realtime applications [5].

In this work, Snort IDS [6], a popular open-source signature based IDS [7], has been used to trigger alerts from the CSE-CIC-IDS2018 [8]. To the best of the author's knowledge, limited work has been done on the CSE-CIC-IDS2018 dataset unlike the CICIDS2017 [9], [1] dataset which is becoming popular. These two datasets have identical attack scenarios; however, the CSE-CIS-IDDS2018 simulates a 10 day attacking period using 50 attack machines, 420 victim machines and 30 servers. The CICIDS2017 dataset was simulated over 5 days, with one day of normal activity, and four days of both attack and normal activity.

The rest of the sections in this paper are organised as follows. In Section II, relevant research work on generating sequential network attacks and evaluating prediction of IDSs is presented. A theoretical background on HMM and related concepts is given in Section III. The experimental methodology which includes the presented HMM design and experimental setup is described in Section IV. In Section V, the results are presented and discussed and, finally, conclusions and future work are given in Section VI.

II. RELATED WORK

Diverse work on HMM and intrusion detection and prediction has been conducted based on outdated datasets. Two recent public datasets, CICIDS2017 [10] and CSE-CIC-IDS2018 [8] are currently available. They contain normal traffic, popular and modern common attack scenarios including heartbleed, Brute-force, Botnet, and Denial of Service (DoS). Despite being publicly available, limited work has been done to utilise these datasets for evaluation, testing and tuning of real-time IDS deployments.

Authors in [11] conducted a comprehensive analysis of the CICIDS2017 dataset. They explored the flaws in this dataset and consequently relabelled it to address the high class imbalance problem. Their focus was on the dataset design for effective implementation in intrusion detection and prediction. In [12], an evaluation of the efficacy of various unsupervised anomaly detection techniques in flagging multiple attack types was performed.

In [13], an adaptive anomaly-based IDS using genetic algorithm and profiling is proposed. Using the CICIDS2017 dataset, a detection and false positive rate of 92.85% and 0.69% respectively was achieved, and the technique succeeded to iteratively adapt to new attacks.

More closely related work is conducted in [14], where a multi-layer HMM IDS is developed. Feature selection and creation is applied on CICIDS2017 dataset, thereafter singular vector decomposition is used on principal component analysis for feature extraction and reduction. Vector quantisation using K-means clustering was then used prior to HMM parameter estimation to create cluster labels.

III. OVERVIEW OF HIDDEN MARKOV MODEL

HMM is a probabilistic ML framework that has two intertwined processes, namely, states and observation. Unlike Markov chains, where states have defined transition probabilities, the states in HMM are concealed, thus the term "hidden". However, the modelled process also generates observations, which can be leveraged to infer the state from which the observations have been emitted, as seen

*This work has been supported by the Gulf Science, Innovation and Knowledge Economy Programme of the UK Government under UK-Gulf Institutional Link grant IL 279339985.

in Fig. 1. The figure demonstrates a left-right model where any observation can be emitted from one or more states. Commonly used notations and definitions for HMM are as follows [5], [14]:

- V: Set of M distinct observation symbols, $V = \{v_1, v_2, ...v_M\}$
- Q: Set of distinct N hidden states, $Q = \{q_1, q_2, ...q_N\}$
- O: Observation sequence, $O = \{o_1, o_2, ..., o_t, ..., o_T\}$
- S: Set of hidden states, $S = \{s_1, s_2, ..., s_t, ...s_T\}$
- A: State transition matrix, $A = \{a_{ij}\}, 1 \leq i \leq N, 1 \leq j \leq N$, where $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$, is the probability of transitioning from state s_i at time t to s_j at time $t+1$
- B: Observation probability distribution, $\{b_j(v_k)\}$, where $b_j(v_k) = P(o_t = v_k | s_t = q_j), 1 \leq k \leq M$, is the probability that symbol v_k is observed in state s_j at time t
- π: Prior state probability distribution, $\pi = \{\pi_i\}$, where $\pi_i = P(s_1 = q_i | t = 1)$, is the initial state probability for state i.

For A, B, and π, each row sums to unity. For real world applications, three fundamental HMM problems that conventionally have to be solved are:

1) *Training or Learning:* This estimates the best parameters that represent the HMM, while maximising $P(O|\lambda)$. Conventionally, Baum Welch (BW) algorithm, an Expectation Maximisation (EM) based approach, specifically formulated for HMM, is used as a de-facto training technique.
2) *Decoding:* This attempts to obtain the most likely state sequence, given an observation sequence, O, and HMM, λ. The Viterbi algorithm is commonly used to address this problem.
3) *Evaluation:* This problem determines the $P(O|\lambda)$ given an observation sequence, O, and model, λ. The forward algorithm is used to compute the confidence of being in any state at time, t. It uses the forward variable, $\alpha_t(i)$, which is the probability of observing a sequence, $O = \{o_1, o_2, ...o_t\}$ and knowing the state $s_t = q_i$, as follows:

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(j) a_{ij} \tag{1}$$

IV. EXPERIMENTAL METHODOLOGY

A. CSE-CIC-IDS2018 Data Processing

The CSE-CIC-IDS2018 dataset, adopted in this work, comprises seven modern attack scenarios over a large network for 10 days. The attack scenarios are Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. To acquire observations from this dataset, Snort version 2.9.11.1 was used to first obtain alerts and later process these alerts before feeding into an HMM. Snort IDS standard configuration and default rules are aggregated using PulledPork 0.7.4 [15], a Perl script that consolidates Snort rules. MATLAB release R2019a was used to extract fields from the alerts file prior to training and evaluation. Table I shows the number of alerts obtained from Snort using the default rules and the alert's corresponding state.

B. Experimental Design and Setup

The various attacks in Table I were aggregated sequentially into six stages of a left-right model. In total, 21 distinct observation types were obtained from Snort IDS and these observations would appear in different states. In each attack stage, the training and evaluation datasets were constructed by including 70% and 30% of the samples, respectively. Training was performed using BW and Viterbi Training (VT) algorithms in MATLAB software. The former, updates all possible paths from each node, whereas the latter updates only the best (Viterbi) path from each node of a trellis.

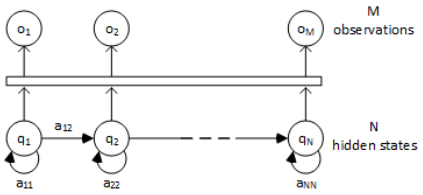

Fig. 1. A left-right HMM process

TABLE I
ALERTS TRIGGERED BY DEFAULT SNORT RULES FOR CSE-CIC-IDS2018 DATASET

Attacks based on pcap files	Number of alerts	HMM state
FTP & SSH Brute-force	282784	1
DoS GoldenEye & Slowloris	8185	2
DoS SlowHTTPTest	8139	2
DoS Hulk	205	2
DDoS LOIC HTTP & UDP	151	3
DDoS LOIC UDP	136	3
DDoS LOIC HOIC	329	3
Brute-force Web &XSS, SQL injection	630	4
Infiltration	4196	5
Botnet	152818	6

Prior to learning of the optimum HMM parameters, both BW and VT techniques require initialisation of parameters. Three common ways of initialisation are uniform, random and count-based methods. Firstly, the uniform initialisation defines each row element of A and B as $1/N$ and $1/M$ respectively. Secondly, the random initialisation generates stochastic numbers from a uniform distribution between 0 and 1. Finally, the count based method [16] uses part of the training dataset to compute B. Each element of B, $b_j(k)$, is obtained by computing the number of occurrences of the observation, v_k, in state j.

A total of 320275 observations out of the 457550 observations extracted from Snort IDS were reserved for training and the remaining were allocated for evaluation. A sample window size ranging from 50 to 1200 with increments of 50 samples, was applied on the evaluation dataset. The analysis is iteratively performed each time shifting the window size along the evaluation samples up to the end of the dataset. In the end of the run of the window size along the evaluation dataset, all generated results are averaged to produce the detection accuracy.

V. RESULTS AND DISCUSSION

The accuracy of determining all states (AS), the current state (CS), and predicting the possible next state (NS) was computed for both BW and VT. Regarding AS and CS determination, the Viterbi decoding was applied directly. NS was predicted by first estimating the most probable observation symbol, v_k, at time $t+1$, while in state j, using Eq. (2) [17].

$$P_{t+1}(v_k) = \sum_{r=1}^{N} a_{jr} b_r(v_k) \tag{2}$$

The estimated symbol is then appended to the known observation sequence before applying Viterbi decoding to recompute the most probable state sequence. The last state of the sequence is then considered as the most probable next state.

There is a comparable performance between AS and CS detection, and only results for CS are discussed in this work. With the

exception of VT-uniform, which had a 3.06% accuracy improvement when increasing the window size from 50 to 100 samples, the rest of the techniques are not significantly affected by increasing the sample window size. Specifically, considering all techniques, the changes in accuracy were trivial with window sizes larger than 150 samples. Thus, a sample window size of 150 was considered. Figure 2 depicts both the detection and prediction accuracy of CS and NS when using BW and VT algorithms.

Regarding AS, it can be observed that count-based initialisation of BW and VT achieves a very high accuracy of about 97%. The uniform initialisation is the second best initialisation technique with about 65% and 61.8% detection accuracy for VT and BW respectively. For NS prediction, it can be observed that the VT-uniform had the highest performance with an average accuracy of 65%.

Contrary to AS and CS prediction, the count-based technique did not perform better than the other techniques, with the VT-count based method completely failing to make a feasible prediction, regardless of any window size. A plausible explanation to this could be that the prediction method as NS prediction relied on next observation prediction. Nevertheless, the Uniform VT can be a better proposed training technique. It can be deduced that a single technique may not scale well in detecting AS, CS and predicting NS.

VI. CONCLUSION AND FUTURE WORK

This paper has presented and evaluated the two conventional training algorithms for HMM, namely, Baum Welch (BW) and Viterbi Training (VT). For both of these algorithms, three standard initialisation techniques, uniform, random and count-based, have been evaluated and their performance discussed. For all scenarios, the performance of the HMM was analysed based on detection of all states (AS), current state (CS) and the next state (NS) given an observation sequence. The experiments have been conducted using the CSE-CIC-IDS20118, a modern dataset comprising seven different attack scenarios over a large network environment.

Results have shown that the count-based initialisation technique outperforms both the uniform and random initialisation when detecting AS and CS. On average, count-based BW and VT have about 97.5% and 97.0% accuracy, respectively, for AS prediction. For CS detection, the performance is comparative to AS detection with a minor drop of about 0.2%. The prediction of NS is around 65% for uniform and random initialisation techniques on both BW and VT.

For all prediction approaches, regardless of the explored training techniques, there is no significant improvement with increasing the window sample size. In practice, the training techniques are deployed by connecting the output of an IDS or a database that stores alerts. These alerts are then fed to an HMM. HMM implementation is scalable and can be applied to any dataset as long as states and observations are clearly defined.

In future, HMM shall be applied on other anomaly detection techniques using the CSE-CIC-IDS2018 dataset, the performance of diverse datasets shall be examined, and a comparative analysis of the computational complexity and success rate of state of the art techniques shall be conducted.

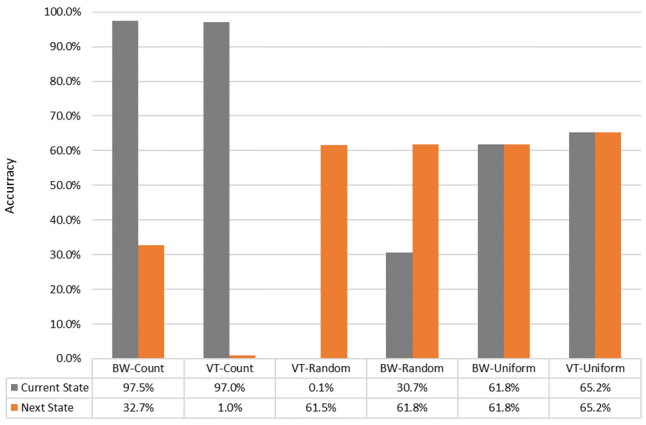

Fig. 2. Current and next state detection accuracy of Baum Welch and Viterbi training using a sample window size of 150.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems and Security and Privacy (ICISSP 2018)*, 2018, pp. 108–116.

[2] S. M. Hussein, "Performance Evaluation of Intrusion Detection System Using Anomaly and Signature Based Algorithms to Reduction False Alarm Rate and Detect Unknown Attacks," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, Dec 2016, pp. 1064–1069.

[3] M. Sato, H. Yamaki, and H. Takakura, "Unknown Attacks Detection Using Feature Extraction from Anomaly-Based IDS Alerts," in *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*. IEEE, Jul 2012, pp. 273–277.

[4] T. Shawly, A. Elghariani, J. Kobes, and A. Ghafoor, "Architectures for Detecting Real-time Multiple Multi-stage Network Attacks Using Hidden Markov Model," *Cornell University Library arXiv*, pp. 1–29, 2018.

[5] A. A. Ramaki, A. Rasoolzadegan, and A. J. Jafari, "A Systematic Review of Intrusion Detection using Hidden Markov Models: Approaches, Applications, and Challenges," *Journal of Modelling in Engineering*, vol. 16, no. 53, pp. 16–16, 2018.

[6] Cisco, "Snort - Network Intrusion Detection and Prevention System," 2018. [Online]. Available: https://www.snort.org, [Accessed: 03-Jul-2019].

[7] K. Salah and A. Kahtani, "Improving Snort performance under Linux," *IET Communications*, vol. 3, no. 12, pp. 1883–1895, 2009.

[8] Canadian Institute of Cybersecurity, "CSE-CIC-IDS2018 on AWS," 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html, [Accessed: 03-Jul-2019].

[9] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Software Networking*, vol. 2017, no. 1, pp. 177–200, Jan 2017.

[10] Canadian Institute of Cybersecurity, "Intrusion Detection Evaluation Dataset (CICIDS2017)," 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html, [Accessed: 03-Jul-2019].

[11] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *International Journal of Engineering & Technology*, vol. 7, no. January, pp. 479–482, 2018.

[12] B. J. Radford, B. D. Richardson, and S. E. Davis, "Sequence aggregation rules for anomaly detection in computer network traffic," *CoRR*, vol. abs/1805.03735, 2018. [Online]. Available: http://arxiv.org/abs/1805.03735

[13] P. A. A. Resende and A. C. Drummond, "Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling," *Security and Privacy*, vol. 1, no. 4, p. e36, Jul 2018.

[14] W. K. Zegeye, R. A. Dean, and F. Moazzami, "Multi-Layer Hidden Markov Model Based Intrusion Detection System," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 265–286, Dec 2018.

[15] J. Cummings, M. Shirk, and PulledPork Team, "Pulledpork," 2017. [Online]. Available: https://github.com/shirkdog/pulledpork, [Accessed: 03-Jul-2019].

[16] N. C. Laan, D. F. Pace, and H. Shatkay, "Initial model selection for the Baum-Welch algorithm as applied to HMMs of DNA sequences," in *First Canadian Student Conference on Biomedical Computing*, Kingston, 2005.

[17] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, "Intrusion alert prediction using a hidden markov model," *CoRR*, vol. abs/1610.07276, 2016. [Online]. Available: http://arxiv.org/abs/1610.07276