Predicting Network Threat Events Using HMM Ensembles*

Akshay Peshave[1], Ashwinkumar Ganesan[1**], and Tim Oates[1]

University of Maryland, Baltimore County, Baltimore MD 21250, USA
{peshave1,gashwin1,oates}@umbc.edu

Abstract. Network traffic analysis, with the objective of identifying and preempting malicious campaigns, is an active area of research. An effective model that predicts future malicious network events based on observed malicious event sequences can aid with preemptive action that includes intervention by a security analyst. Predicting threat events that are part of a cybersecurity threat campaign that spans a long duration of time remains a challenge as the time lag between various steps in a campaign is unbounded. In this paper, we describe an approach to create an ensemble of Hidden Markov Models trained on sequences of malicious network events. The ensemble is used to predict the next expected malicious event given an already observed malicious traffic sequence at any network host. Ensembles of different sizes in combination with two prediction strategies are evaluated using prediction accuracy relative to two baselines predictors.

Keywords: Network Threat Prediction · Malicious Traffic Sequence Analysis · Hidden Markov Model Ensemble.

1 Introduction

Network traffic analysis with the objective of identifying malicious campaigns is an active area of research. Network threats escalate as the severity of malicious or adverse events increases based on threat or severity levels defined by network monitoring tools. The ability to preempt network threat escalations and proactively address the situation is a current need. Several network monitoring tools [10,9] are available for monitoring network traffic at different granularity. These sensors utilize large and diverse rule sets to identify malicious network events, and in some cases long, persistent patterns of malicious network traffic. As organization networks have grown in size and complexity, the available attack surfaces (i.e. potential vulnerabilities in network security setups and available avenues to compromise a network) have increased. This coupled with sophisticated and persistent network threat campaigns has led to increasingly complex network traffic monitoring that generates enormous amounts of data.

* This research was conducted in the UMBC Accelerated Cognitive Computing Lab (ACCL), supported in part by a gift from IBM.

** Research completed prior to joining Amazon.

B. Li et al. (Eds.): Advanced Data Mining and Applications (ADMA) 2021, LNAI 13087, pp. 229–240, 2022.
The final publication is available at Springer via https://doi.org/10.1007/978-3-030-95405-5_17.

Analyzing such large amounts of raw information to generate actionable insights has become increasingly difficult for human analysts. Human-in-the-loop paradigms for such analyses that are facilitated by machine learning based systems have been evolving over time. These machine learning based systems employ sophisticated pattern analysis and anomaly detection methods to provide actionable insights to human analysts.

We propose an approach to train an ensemble of Hidden Markov Models (HMMs) to predict future malicious network events based on observed malicious event sequences. We evaluate two prediction strategies using the trained HMM ensemble and compare their prediction accuracy to two baseline predictors, namely the uniform random predictor and the most frequent event predictor.

The rest of the paper is organized as follows: Section 2 discusses the background and related research work. Section 3 describes the proposed HMM ensemble, its training and the prediction strategies employed by the ensemble. Section 4 describes the data set used to evaluate our method followed by Section 5 that discusses the performance of our approach relative to the baselines. Section 6 concludes this work.

2 Background & Related Work

Machine learning applications in cybersecurity event analysis for intrusion pattern characterization and malicious traffic prediction is an active area of research [3,7]. HMMs have been shown to be effective for cybersecurity sequence analysis tasks. HMMs have been utilized in a competitive ensemble setting for detecting specific intrusion methods such as SQL-injection [2]. Pre-configured HMMs used for intrusion detection [11] are highly sensitive to the selected feature set.

An anomaly detection approach for detecting cyber-attacks [13] trains a Markov chain model using past observed data in normal operating conditions at a host or on the network as a whole. The model assesses the generation probability of data sequences in live conditions to detect anomalies when the generation probability is lower than expected. This model is sensitive to noise in the training data. Further, novel benign traffic patterns in live conditions may be detected as anomalies.

A stochastic prediction model that learns absorbing Markov chains from simulated attack graphs [1] is another approach for network threat prediction. It is based on a cyber-situational awareness model that describes four levels of increasing awareness in a cybersecurity ecosystem - perception, comprehension, mitigation and forecasting. Forecasting requires effective perception and comprehension of network traffic to identify malicious sequences that need mitigation.

Intrusion Detection Systems (IDS) and network sensor alerts are typically fine-grained and only a very small subset of alerts may constitute absorbing states in the context of a complete attack campaign. Further, threat escalation prediction implies the ability to preempt escalation from sequences of varied

lengths available in the training (or observed) traffic data. These sequences may or may not contain the predefined absorbing states required by the absorbing Markov chains described in [1].

Our proposed approach trains an HMM ensemble using observed malicious event sequences of varied lengths without explicitly specifying absorbing states. Further, benign network traffic is not utilized. This ensures that our method is not sensitive to evolving patterns of benign traffic and relies solely on malicious events described in, and consequently detected by, IDS and network monitoring tools.

3 Ensemble of Hidden Markov Models

The goal of this work is to build a model to predict the next malicious network event given a sequence of observed malicious network events. The set of malicious events are predefined events in the network monitoring tools deployed on the network (like snort [10]) that are considered as sensors. The model is trained using malicious event sequences observed at each network host as detected by the network sensors. The working assumption is that a network monitoring tool will assign a single class or category to each malicious network traffic event it detects, i.e. the tool does not provide a probabilistic set of malicious event classes or categories.

Section 3.1 discusses preliminaries about ergodic Hidden Markov Models and describes the structure of HMMs used for our proposed method along with related nuances. Section 3.2 describes clustering of malicious event sequences using HMM clustering. Lastly, Section 3.3 describes the creation of the HMM ensemble using clustered sequences and two malicious network event prediction strategies using the ensemble that will be evaluated in this work.

3.1 Hidden Markov Model Structure

HMM Preliminaries : HMMs learn patterns from a set of observed sequences and model them as a set of latent/hidden states, inducing probabilities of transitioning between the states and probability distributions over possible emission symbols at each state. An HMM trained on a set of sequences models the state transitions and per-state emission probability distributions that are most likely to generate those sequences. An HMM can be formalized using the following properties:

$N =$ number of states
$T =$ number of emission symbols
$s_i =$ hidden state, $1 \leq i \leq N$
$y_j =$ emission symbol, $1 \leq j \leq T$
$\boldsymbol{\varphi} = N$xN matrix, where $\boldsymbol{\varphi}_{i,j}$ is the transition probability $P(s_j|s_i)$
$\boldsymbol{\theta} = N$xT matrix, where $\boldsymbol{\theta}_{n,t}$ is the emission probability $P(y_t|s_n)$

Figure 1 shows the general structure of an ergodic HMM using the symbols described above. An ergodic HMM is one in which every hidden state can transition to all the HMM hidden states including itself.

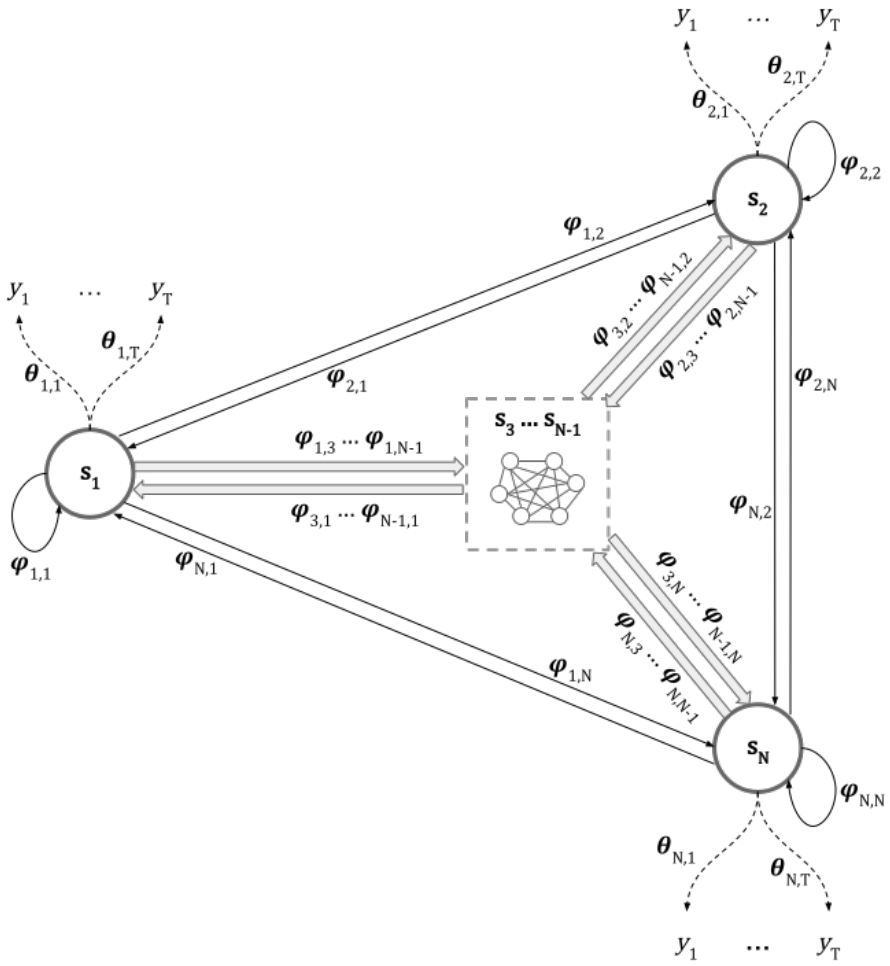

Fig. 1. Ergodic Hidden Markov Model Structure. All states can transition to any of the other states including itself. Additionally, each hidden state may emit all of the emission symbols.

HMM Structure for the Proposed Method : The threat pyramid for advanced persistent network threats [4] consists of 7 categories of malicious network traffic or events. These 7 categories can be further abstracted into 3 threat

levels: reconnaissance, infiltration and exfiltration. Reconnaissance activities encompass attempts to understand the network structure and identify potential network vulnerabilities. Infiltration network activities focus on exploiting discovered vulnerabilities to gain access to network hosts and/or organization data. Exfiltration refers to network activities aimed at exfiltrating accessed data from the organisation's network and is possible once the attacker successfully infiltrates the network.

The proposed method uses ergodic Hidden Markov Models with $N = 3$, in which each state may emit any of the emission symbols that correspond to the dictionary of malicious events provided by the network sensors. The choice of $N = 3$ is based on the 3 abstract network threat levels of reconnaissance, infiltration and exfiltration discussed above.

HMMs trained on malicious event sequences that contain events belonging to all three threat levels will induce a model that generates macro-patterns of malicious event sequences that result in ex-filtration activities. On the other hand, HMMs trained on sequences that do not contain events belonging to all three threat levels will induce a model that generates micro-patterns of malicious event sequences belonging to ≤ 2 threat levels.

Discount Smoothing Applied to HMMs. It is expected that every malicious event sequence will not contain all the emission symbols from the event dictionary. Further, some emission symbols may occur with higher frequency than others in a sequence. This is due to two reasons:

1. Most network intrusion campaigns may go through prolonged periods of reconnaissance before they find exploitable vulnerabilities that allow them to infiltrate the network and exfiltrate data [1]. Consequently, frequencies of observed malicious events are unbalanced and often heavily skewed towards reconnaissance events.
2. Sequences that go through one or two threat levels may have a skew in frequencies of events belonging to each threat level as well as a skew in frequencies of events belonging to the same threat level.

Thus, a large portion of emission symbols may have zero emission probabilities at one or more states in an HMM. Smoothing of the emission probabilities for each hidden state in the HMM helps eliminate the zero emission probabilities of some state emission symbols by assigning tiny emission probabilities to them. We use a variation of discount smoothing that adds a tiny probability mass to near-zero ($\leq 10^{-4}$) emission probabilities of a state. This probability mass is obtained by discounting it from the other emission probabilities at that state.

Let, $\boldsymbol{A} = \{a \in [1, T]\}$ be indices of emission symbols at state s_i such that $\forall a \in \boldsymbol{A}, \boldsymbol{\theta}_{i,a} \leq 10^{-4}$. Let, $\boldsymbol{B} = \{b \in [1, T]\}$ be indices of emission symbols at state s_i such that $\forall b \in \boldsymbol{B}, \boldsymbol{\theta}_{i,b} > 10^{-4}$. The discount d_i at state s_i is computed as follows:

$$d_i = \sum^{b \in \boldsymbol{B}} 10^{-3} * \theta_{i,b} \tag{1}$$

This discount is evenly distributed among the emission probabilities of emission symbols with indices in $\boldsymbol{A}$ while subtracting the probability mass contributed by each emission symbol in $\boldsymbol{B}$ from their respective emission probabilities. The resultant emission probabilities at state s_i after discount smoothing are as follows:

$$\forall a \in \boldsymbol{A}, \quad \boldsymbol{\theta}_{i,a} = \boldsymbol{\theta}_{i,a} + \frac{d_i}{|A|}$$
$$\forall b \in \boldsymbol{B}, \quad \boldsymbol{\theta}_{i,b} = \boldsymbol{\theta}_{i,b} - 10^{-3} * \theta_{i,b}$$

3.2 Event Sequence Clustering

Clustering of sequences is a hard problem. Traditional methods to cluster categorical data such as k-modes [5,6] can be employed by treating event sequences as event itemsets. But, this approach disregards temporal ordering of the events in the sequence. In the case of malicious network event sequences, both the events occurring in the sequence as well as their order of occurrence is important. We need a clustering method which takes both these into account.

An approach to cluster sequences trains one HMM per sequence [12]. This work shows that clustering per-sequence HMMs using pairwise KL-divergence is equivalent to clustering of the individual sequences that the HMMs are trained on. [12] explain that this clustering approach is computationally expensive and clustering over a representative sample of sequences is a recommended practice.

In our method, per-sequence HMMs are clustered using symmetric KL–divergence over sub-sequence generation probabilities of the HMMs as the distance metric for hierarchical, agglomerative clustering [8]. The linkages between clusters are evaluated in order to merge clusters hierarchically. The flattened clusters of HMMs are obtained by applying a threshold on the cophenetic distances between training sub-sequences. We utilize this sub-sequence clustering approach in the first stage of our method.

3.3 Ensemble Creation and Prediction Methods

The ensemble of HMMs consists of a set of one cluster-HMM per sub-sequence cluster obtained above. These HMMs have the same structure as described in Section 3.1. The intuition is that cluster-HMMs will enable good prediction performance and generalization over a wide range of sub-sequence variations using a small sized ensemble. The prediction from each cluster-HMM is the most probable emission from the most probable next state in the HMM having observed the input sequence of malicious events. The Viterbi process is used to reach the current state in the HMM by observing the input sequence. The most probable next malicious event can be obtained from the emission probability distribution of the current state.

Each cluster-HMM can provide a malicious event prediction given an observed event sequence. Two methodologies are assessed in this work for the HMM ensemble to collectively predict the next malicious event:

1. **Majority Vote Prediction**: This strategy generates next-step emissions as predictions from all HMMs in the ensemble given an input subsequence. Each prediction is given a uniform weight of 1. The final prediction by the ensemble is the prediction with the most cumulative weight i.e. the most number of votes.
2. **Maximum Generation Likelihood Prediction**: This prediction strategy chooses the HMM from the ensemble that has the maximum generation likelihood for the input subsequence and uses this HMMs next-step emission prediction as the final prediction by the ensemble.

4 Data Set

The dataset used for the evaluation of our approach is Snort logs generated for the U.S. National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC)[1] 2012 capture files. These logs were obtained from the SecRepo[2] website. The MACCDC competition has two competing teams and consists of two rounds. Each competing team gets an opportunity to be the attacking team while the other team defends the network setup for the competition. In the first round, team A attacks the network while team B defends it. In the second round, team B attacks the network while team A defends it. Each team employs their own campaigns to attack the network and circumvent preemptive and protective strategies implemented by the defending team.

The network staged for the competition consists of 4,757 hosts of which 4,739 face some incoming malicious network traffic events during the entire competition, as evidenced from the dataset. The malicious traffic generates 2,677,375 snort alerts distributed over 25 snort classes over the entire course of the competition as shown in Table 1. The 4,739 malicious network event trajectories, one per sink host on the network, are used to generate sub-sequences of length 10 using a sliding event index window. This results in 658,343 sub-sequences of which 51,606 are unique. The evaluation in Section 5 samples these sub-sequences to train and test the HMM ensemble for malicious event prediction using our approach.

5 Evaluation

The evaluation of the HMM ensemble prediction performance needs us to sample the 51,606 unique sub-sequences generated as described in Section 4. We train the HMM ensemble using three random samples of sizes 100, 1,000 and 5,000. The cophenetic distance threshold for agglomerative clustering is specified as a percentile of the all the pairwise cophenetic distances between the HMMs trained on the sub-sequence samples. As the sample size grows the separation of HMMs into multiple clusters is achieved at a lower percentile. This can be seen in Table 2.

[1] http://www.maccdc.org
[2] http://www.secrepo.com

Table 1. Dataset Description. The table shows the distribution of observed malicious activity alerts over different Snort alert classes. As seen in the table, the dataset is skewed wherein Snort classes such as *"Attempt to Login By a Default Username and Password"* and *"Detection of a Network Scan"* are observed less that 100 times whereas *"Web Application Attack"* is observed $1,348,695$ times.

ID	SNORT Alert Class	Total	Team A	Team B
1	A Network Trojan was Detected	3272	2471	801
2	A Suspicious Filename was Detected	7	7	0
3	A Suspicious String was Detected	192	182	10
4	Access to a Potentially Vulnerable Web Application	31,376	25,895	5481
5	Attempt to Login By a Default Username and Password	55	55	0
6	Attempted Administrator Privilege Gain	10,014	9777	237
7	Attempted Denial of Service	469	218	251
8	Attempted Information Leak	421,782	244,177	177,605
9	Attempted User Privilege Gain	2835	2675	160
10	Decode of an RPC Query	674	396	278
11	Detection of a Denial of Service Attack	264	250	14
12	Detection of a Network Scan	34	24	10
13	Detection of a Non-Standard Protocol or Event	207	199	8
14	Executable Code was Detected	9301	5139	4162
15	Generic Protocol Command Decode	27,259	25,566	1693
16	Information Leak	8672	4674	3998
17	Misc activity	588,532	225,353	363,179
18	Misc Attack	19,649	11,269	8380
19	Potential Corporate Privacy Violation	148,860	141,401	7459
20	Potentially Bad Traffic	36,995	23,752	13,243
21	Successful Administrator Privilege Gain	38	11	27
22	Successful User Privilege Gain	7003	35	6968
23	Unknown Traffic	39	16	23
24	Unsuccessful User Privilege Gain	11,151	11,121	30
25	Web Application Attack	1,348,695	1,337,473	11,222
	Total	2,677,375	2,072,136	605,239

Further, the number of clusters formed does not vary as we continue decreasing the cophenetic distance threshold after the initial breakup into multiple clusters is achieved. This is indicative of optimal discrimination achieved between the generative process of the sub-sequence HMMs at the cophenetic distance threshold at which the first split into multiple clusters occurs.

We use two baselines for comparison of prediction accuracy by the HMM ensemble. The first baseline is the uniform random predictor accuracy. This is the accuracy of a predictor that randomly predicts one of the 25 snort alert class with uniform probability. The random predictor accuracy is $\frac{1}{25}$ or 4%. The second baseline is the most frequent class predictor accuracy. This is the accuracy

Table 2. Size of HMM Ensembles defined by Cophenetic Distance Linkages for differing thresholds and training sample size.

Linkage Threshold %ile	**Number of Clusters Per Training Sample Size**		
	100	1,000	5,000
0.010	7	528	2,559
0.015	7	528	2,559
0.020	7	528	1
0.025	7	528	1
0.050	7	528	1
0.075	7	528	1
0.080	7	528	1
0.090	7	528	1
0.100	7	528	1
0.200	7	1	1
0.300	7	1	1
0.400	7	1	1
0.500	7	1	1
1.000	7	1	1
2.000	1	1	1
3.000	1	1	1
4.000	1	1	1
5.000	1	1	1

of a predictor that always predicts the most frequently observed snort alert class in the dataset. This most frequent class predictor accuracy is 21.98%.

Table 3. Accuracy of Prediction Strategies for 25% Random Sample of MACCDC 2012 Subsequences as Test Set.

Training Sample	**Prediction Strategy**	
	Majority Vote	**Generation Likelihood**
100	58.63	7.62
1,000	55.25	51.12
5,000	55.22	51.20

The evaluation is done using two test sets and prediction accuracy is measured for both the prediction strategies described in Section 3. The first test set is a random sample of 25% of the unique sub-sequences from the dataset. The second test set is the complete set of unique sub-sequences from the dataset inclusive of the training sample. Inclusion of the training random sample does not bias the test set accuracy in the latter case since the training set is at most 5,000 sub-sequences which is less than 10% of the complete dataset of unique sub-sequences.

The prediction accuracy of both prediction strategies of the HMM ensemble for the 25% random sample test set is shown in Table 3. Prediction accuracy for the majority vote prediction strategy for all three samples is similar and outperforms both the baseline predictors. The maximum generation likelihood prediction strategy under-performs both the baseline predictors when trained using 100 samples while outperforming both the baselines when trained using 1,000 and 5,000 samples.

Table 4. Accuracy of Prediction Strategies for All MACCDC 2012 Subsequences as Test Set.

Training Sample	**Prediction Strategy**	
	Majority Vote	**Generation Likelihood**
100	47.42%	2.10%
1000	54.82%	14.91%
5000	54.78%	14.95%

The prediction accuracy of both prediction strategies of the HMM ensemble for the complete dataset as the test set is shown in Table 4. Prediction accuracy for the majority vote prediction strategy when trained using 1,000 and 5,000 samples is similar while its slightly lower when trained using 100 samples. This strategy outperforms the two baselines for all three training sample sizes. Training using the larger sample sizes generalizes to the complete dataset better than training using the smaller sample size.

Prediction accuracy for the maximum generation likelihood prediction strategy using the complete dataset as the test set is lower than its accuracy for the 25% random sample test set. This strategy under-performs the most frequent class predictor contrary to its performance for the 25% random sample test set. This is suggestive of the lack of generalization of the maximum generation likelihood prediction strategy. This is not the case with the majority vote prediction strategy when trained using 1,000 and 5,000 random samples. The prediction accuracy for both these random sample sizes for this strategy does not vary much between the two test sets. The under-performance and lack of generalization of the maximum generation likelihood prediction strategy also indicates that it is uncommon to find an HMM in the ensemble that can predict the next malicious network event by itself. Hence, an ensemble of HMMs serves our prediction task better.

6 Conclusion

An ensemble of Hidden Markov Models trained using clustered malicious network event sequences performs significantly better than the two baseline predictors - uniform probability random predictor and most frequent label predictor. The majority vote ensemble prediction strategy outperforms the maximum generation likelihood prediction strategy for all three training sample sizes used to

train the HMM ensemble. The majority vote HMM ensemble prediction strategy generalizes better than the latter for prediction over the entire dataset using training sample sets that are approximately 2% and 10% of the dataset.

References

1. Abraham, S., Nair, S.: Cyber security analytics: A stochastic model for security quantification using absorbing markov chains. In: Journal of Communications. vol. 9, pp. 899–907 (12 2014)
2. Ariu, D., Tronci, R., Giacinto, G.: Hmmpayl: An intrusion detection system based on hidden markov models. Computers I& Security **30**(4), 221 – 241 (2011). https://doi.org/https://doi.org/10.1016/j.cose.2010.12.004, `http://www.sciencedirect.com/science/article/pii/S0167404811000022`
3. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys Tutorials **18**(2), 1153–1176 (Secondquarter 2016). https://doi.org/10.1109/COMST.2015.2494502
4. Giura, P., Wang, W.: A context-based detection framework for advanced persistent threats. In: 2012 International Conference on Cyber Security. pp. 69–74 (2012). https://doi.org/10.1109/CyberSecurity.2012.16
5. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: In The First Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 21–34 (1997)
6. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. In: Data Mining and Knowledge Discovery 2(3). pp. 283–304 (1998)
7. Husák, M., Komárková, J., Bou-Harb, E., Čeleda, P.: Survey of attack projection, prediction, and forecasting in cyber security. IEEE Communications Surveys Tutorials **21**(1), 640–660 (2019). https://doi.org/10.1109/COMST.2018.2871866
8. Müllner, D.: Modern hierarchical, agglomerative clustering algorithms. arXiv e-prints arXiv:1109.2378 (Sep 2011)
9. Paxson, V.: Bro: A system for detecting network intruders in real-time. In: Proceedings of the 7th Conference on USENIX Security Symposium - Volume 7. p. 3. SSYM'98, USENIX Association, USA (1998)
10. Roesch, M.: Snort - lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX Conference on System Administration. p. 229–238. LISA '99, USENIX Association, USA (1999)
11. S. Joshi, S., Phoha, V.: Investigating hidden markov models capabilities in anomaly detection. In: Proceedings of the Annual Southeast Conference. vol. 1, pp. 98–103 (01 2005)
12. Tamang, S., Parsons, S.: Using semi-parametric clustering applied to electronic health record time series data. In: Proceedings of the 2011 Workshop on Data Mining for Medicine and Healthcare. pp. 72–75. DMMH '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/2023582.2023596, `http://doi.acm.org/10.1145/2023582.2023596`
13. Ye, N., Zhang, Y., Borror, C.M.: Robustness of the markov-chain model for cyber-attack detection. IEEE Transactions on Reliability **53**(1), 116–123 (March 2004). https://doi.org/10.1109/TR.2004.823851