Viterbi Algorithm for Intrusion Type Identification in Anomaly Detection System

january 14th 2019

Context

Intrusion Type

- Buffer overflow
  - xlock vulnerability
  - lpset vulnerability
  - kcms_sparc vulnerability
- S/W security vulnerability
- Setup vulnerability
- Denial of service

Markov Chain

A markov Chain is defined by :

. S, A finite set of N states

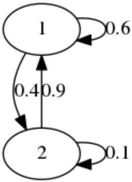
. π, A vector of initial probabilities over S :

$$\pi_i = P(S_1 = i), 1 \leq i \leq N$$

. A, A matrix of probabilities of transitions over SxS :

$$a_{ij} = P(S_t = j | S_{t-1} = i), 1 \leq i \leq N$$

. Markov assumption : $P(S_t | S_{t-1}, S_{t-2}, \ldots, S_1) = P(S_t | S_{t-1})$



$$A = \begin{pmatrix} 0.6 & 0.4 \\ 0.9 & 0.1 \end{pmatrix}$$

Figure: Simple example of Markov Chain

HMM - Hidden Markov Model

– Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.

HMM - Hidden Markov Model

- Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.
- Hidden Markov Model can be viewed as a Bayesian Network

HMM - Hidden Markov Model

- Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.
- Hidden Markov Model can be viewed as a Bayesian Network
- We define a HMM including :

  V, A finite set of M observations

  B, A a matrix of probabilities of observations over state :

$$b_i(k) = P(0_t = V_k | S_t = i)$$

HMM - Forward Algorithm

input : λ The model, O Observed sequence
output : $P(0|\lambda)$
Step 1, Initialization : $\forall i, \alpha_1(i) = \pi_1 b_i(0_1)$
Step 2, Induction :
for $t \leftarrow 2 : T$ **do**

$$\alpha_t = \left[\sum_{j=1}^{N} \alpha_{t-1} q_{ij}\right] b_j O_t$$

end
Step 3, Termination : $P(0|\lambda) = \sum \alpha_t(i)$

1

[1] L. R. Rabiner (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286

HMM - Viterbi Algorithm

input : Λ All the models, O Observed sequence
output : $\arg\max_{\lambda \in \Lambda} P(0|\lambda)$

Step 1, Initialization :
for $i \leftarrow 1 : N$ **do**
  $\delta_1(i) = \pi_i b_i(0_1)$
  $\psi_1(i) = 0$
end
Step 2, Recursion :
for $t \leftarrow 2 : T$ **do**
  for $j \leftarrow 1 : N$ **do**
    $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$
    $\psi_t(j) = \arg\max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$
  end
end
Step 3, Termination :
$P^* = \max_{s \in S} [\delta_T(s)$
$S_T^* = \arg\max_{s \in S} [\delta_T(s)$
Step 4, Backtracking :
for $t \leftarrow T-1 : 1$ **do**
  $S_t^* = \psi_{t+1}(s_{t+1}^*)$
end
return S^*

[2]

[2] A. Viterbi (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2, pp. 260–269

Normal Behaviour Modeling

Normal Behaviour is modelised by a left-to-right HMM λ.

The forward allgorithm is used to decide whether normal or not with a threshold.

Intrusion Detection

Initialization

Show Example

Intrusion Detection

Induction

Show Example

Intrusion Detection

Termination

Show Example

Intrusion Detection

Decision

if $log(P(0|\lambda)) >$ *thresold* **then**
  return *Normal Behaviour*
else
  return *Intrusion*
end

Show Example

Intrusion Detection

Results

Table: The performance of HMM-based IDS. Best results are in bold

Length	Thresold	Detection Rate	F-P Error
10	-9.43	100%	2.626
15	-9.43	100%	3.614
10	-14.42	100%	1.366
15	-14.42	100%	2.718
10	-16.94	100%	0.789
15	-16.94	100%	2.618
10	-18.35	100%	0.553
15	-18.35	100%	2.535
10	-19.63	100%	0.476
15	-19.63	100%	2.508
10	**-20.83**	**100%**	**0.372**
15	-20.83	100%	2.473

Intrusion Type Identification

Process in two steps :

- Viterbi algorithm used to find the optimal state sequence
- Euclidiant distance to identify the intrusion type with the optimal state sequence

Intrusion Type Identification

Initialization

Show Example

Intrusion Type Identification

Recursion

Show Example

Intrusion Type Identification

Termination

Show Example

Intrusion Type Identification

Backtracking

Show Example

Intrusion Type Identification

Decision

Show Example

Intrusion Type Identification

Results

Table: The performance of Viterbi-based Intrusion Type Identification

Attack	Trial	Correct	Incorrect	Rate
Buffer Overflow	20	18	2	90%
Denial of Service	25	9	16	36%
Buffer Overflow	45	27	18	60%

Limitations & Remarks

Try other distance metrics for Intrusion Type Identification : Ja-Min Koo and Sung-Bae Cho (2005). "Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System". In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg

Limitations & Remarks

Try other distance metrics for Intrusion Type Identification : Ja-Min Koo and Sung-Bae Cho (2005). "Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System". In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg

Bad results for Denial of Service : W. Bongiovanni et al. (2015). "Viterbi algorithm for detecting DDoS attacks". In: *2015 IEEE 40th Conference on Local Computer Networks (LCN)*

Limitations & Remarks

- Try other distance metrics for Intrusion Type Identification : Ja-Min Koo and Sung-Bae Cho (2005). "Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System". In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg
- Bad results for Denial of Service : W. Bongiovanni et al. (2015). "Viterbi algorithm for detecting DDoS attacks". In: *2015 IEEE 40th Conference on Local Computer Networks (LCN)*
- No baseline to compare results with other methods

Limitations & Remarks

Baseline