Feature Engineering Method Using Double-Layer Hidden Markov Model for Insider Threat Detection

Xiaoyun Ye , Sung-Sam Hong , and Myung-Mook Han

Department of Computer Science, Gachon University, Seongnam, Korea

Ijfis

Abstract

In the past, most Hidden Markov models based on time series only used the original HMM model. The single-layer models (HMMs) structure has a big problem, and it isn't straightforward to play its due role when it is necessary to make fine adjustments to the scene. So it was impossible to entirely and flexibly perform user behavior. This paper performs feature extraction and analysis of user behavior data of time series. The data labels should be added after the parameters obtained by statistical methods for clustering to obtain the first hidden state, and the layers are further layered according to working hours and outside working hours. The experimental results show that the method has strong applicability and flexibility, and can quickly detect abnormal behavior.

Keywords: Hidden Markov Model (HMM), User behavior, Insider threat, Feature engineering, Anomaly detection.

1. Introduction

The rapidly developing network technology has not only changed the human lifestyle and brought great development opportunities, but also brought more significant challenges in terms of security. After the Edward Snowden incident, people are paying more and more attention to the security of privacy. Cyber-attacks have always existed, but in recent years, the losses caused by internal security risks have become increasingly dangerous. According to the IT Security Risks Survey conducted by Kaspersky Lab and B2B International, 73% of companies have been affected by both intentional and unintentional internal information security incidents. Out of those, a fifth (21%) of companies also lost valuable data that subsequently affected their business [1]. If we do not stop the insider threats, that high-risk situation should be more and more. But the insider threats can't easily be considered as a data-driven problem. We can't use the methods just like an external attack because the complexity of internal attacks is much higher than the outside's attacks. Every employee's daily work is different. The challenge of insider threat detection is to set up the profile of each employee's behavior. Insider threat detection needs to focus on differentiating suspicious users from the other, but we can't directly specify an employee is an attacker. So, we need as many parameters as possible to define the attack's behavior. The hidden Markov model (HMM) is a statistical model used for representing probability distributions over sequences of observations [2]. This model has been used in several domains, such as speech recognition [3], text understanding [4], image

Received: Dec. 31, 2019

Revised : Feb. 24, 2020

Accepted: Feb. 28, 2020

Correspondence to: Myung-Mook Han

(mmhan@gachon.ac.kr)

©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

image identification [5], and microbiology [6]. HMM has also been utilized in [7, 8] to train models using observed network traffic under normal network conditions and to detect diverting sequences of traffic observations. According to the paper by Kabir et al. [9], it shows that double-layer HMM can be used to build models in behavioral data. However, in the face of more important and comprehensive behavioral data, small-scale data models may not achieve better results. For multi-scenario situations, there are disadvantages that cannot be handled. The article only uses all the behaviors in the three rooms for statistics, but if the model is placed in a community, the prediction results will be produced unpredictable because of different human habits. The first layer of the double-layer HMM proposed in this paper is obtained by clustering using an unsupervised algorithm. Assuming that the user's behavior is regular and can be analyzed, density-based spatial clustering of applications with noise (DBSCAN) is a useful division method. The Markov model needs to satisfy two points: continuous-time and discrete events [10–13]. So, Carnegie Mellon University posted an intrusion detection dataset called “CERT” dataset [14]. This dataset satisfies the necessary conditions of the Markov Model. Then we can see the result in [15].

However, the disadvantage of this thesis [15, 16] is that the distinction between normal behavior and abnormal behavior is not clear enough.

So, our contribution in this paper is: (1) a new double-layer HMM model for identifying abnormal behavior, and (2) new feature extraction and modeling method using the hour system. The remainder of this paper is structured as follows: Section 2 presents the description of our related works. We will introduce the feature extraction model and the detection model in detail in Section 3. Section 4 presents the experimental results. Finally, Section 5 gives the concludes and our future work.

2. Related Works

2.1 Hidden Markov Model

An HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states [2]. A Markov model can simple to understand as defining the probability of each behavior occurring and then identifying the product of the possibilities of all actions arising over a period. A simple Markov model's structure is shown in Figure 1. By comparison, when the sequence is relatively long, it isn't straightforward for the Markov chain to distinguish several behaviors. Because HMM has a hidden

Figure 1. Markov model.

layer, it is easier to distinguish between normal and abnormal behavior than the Markov chain. The structure of HMM is shown in Figure 2.

An HMM can be considered as the most straightforward dynamic Bayesian network. The probabilistic parameters of an HMM are shown in Figure 3. We use an example to explain the process of HMM, just like the structure in Figure 4. There are three boxes: blue, orange, and green. And there are five types of balls: A, B, C, D, E. We can get the parameters like the follows:

States = (Blue, Orange, Green)

Observations = (A, B, C, D, E)

Start probability (π) =

Blue: 0.2, Orange: 0.3, Green: 0.5

Transition probability (A) =

Blue: (Blue: 0.3, Orange: 0.6, Green: 0.1),

Orange: (Blue: 0.3, Orange: 0.4, Green: 0.3),

Green: (Blue: 0.2, Orange: 0.7, Green: 0.1)

Observation probability (B) =

Blue: (A: 0.1, B: 0.4, C: 0.5),

Orange: (A: 0.2, B: 0.1, C: 0.7),

Green: (B: 0.5, E: 0.5)

According to the timing, we randomly picked the box and randomly picked the ball inside the box like Figure 5. The first time, we got the blue box and the ‘C’ ball in the box. And next, we got a blue, orange, green box and the ‘A,’ ‘B,’ ‘E’ balls in their boxes.

So, this condition's probability is $(0.2 \times 0.2) \times (0.3 \times 0.2) \times (0.6 \times 0.1) \times (0.3 \times 0.5)$. A complete HMM model requires a set of parameters: $[A, B, \pi]$. A is a transition probability matrix, B is the observation probability matrix, π is the initial state probability. When the feature set is extracted from the dataset, the feature space is obtained.

Figure 2. Hidden Markov model.

Figure 3. Probabilistic parameters of an HMM.

Figure 4. HMM simple.

2.2 Double-Layer HMM

According to the paper by Kabir et al., The feasibility of using double-layer HMM in behavioral data can be shown, but in the face of more extensive and more comprehensive behavioral data, small-scale data models may not be able to obtain better results. For multi-scenario situations, There are obvious disadvantages of insufficient processing. The article only uses all the behaviors in the three rooms for statistics, but if the model is placed in a block or a community, the prediction results will be produced because of different human habits. There is a significant deviation. The first layer of the double-layer HMM proposed in this paper is obtained by clustering using an unsupervised algorithm. Assuming that the user's behavior is regular and can be analyzed, clustering is a useful division method. Double-layer HMM is shown in Figure 5.

Figure 5. Double-layer HMM.

2.3 Viterbi Algorithm

The Viterbi algorithm is named after Andrew Viterbi, which was proposed in 1967. It is a decoding algorithm for convolutional codes on noisy digital communication links [17]. However, it has a history of multiple inventions with at least seven independent discoveries, including those by Viterbi, Nederman, and Winsch, and Wagner and Fischer [18]. “Viterbi path” and “Viterbi algorithm” have become standard terms for applying dynamic programming algorithms to problems involving probability maximization [17].

For example, we got an action shown in Figure 6: B to E. Depending on the situation, we can get three results.

(1) B from blue box.

$$P_1 = \pi_{Blue} * B_{Blue_B} * A_{BlueGreen} * B_{Green_E} \quad (1)$$

(2) B from orange box.

$$P_2 = \pi_{Orange} * B_{Orange_B} * A_{OrangeGreen} * B_{Green_E} \quad (2)$$

Figure 6. HMM simple B to E.

(3) B from green box.

$$P_3 = \pi_{Green} * B_{GreenB} * A_{GreenGreen} * B_{GreenE}. \quad (3)$$

So, we can calculate the probability as:

$$P_1 = 0.2 * 0.4 * 0.1 * 0.5 = 0.004, \quad (4)$$

$$P_2 = 0.3 * 0.1 * 0.3 * 0.5 = 0.0045, \quad (5)$$

$$P_3 = 0.5 * 0.5 * 0.1 * 0.5 = 0.0125. \quad (6)$$

The Viterbi can choose the biggest probability for the model. So, the algorithm should choose as the probability of the action B to E.

2.4 DBSCAN

DBSCAN is a data clustering algorithm proposed by Ester, et al. [19] in 1996. The method used by DBSCAN is straightforward. It arbitrarily selects a core object without a category as a seed and then finds all the sample sets whose core objects can reach a density, which is a cluster. Then continue to select another core object without a category to find a sample set with a dense density to obtain another clustering cluster. Run until all core objects have categories.

It can be easily seen from Figure 7. that the above definition is understood. In Figure 7, MinPts = 5, the red points are the core objects because their ϵ -neighborhood has at least 5 samples. The black samples are non-core objects. All samples with linear core object density are in the hypersphere with the red core object as the center. If they are not in the hypersphere, they cannot be directly dense. The core objects connected by the green arrows in Figure 7. form a sample sequence with a density. All samples in the ϵ -neighborhood of these dense sample sequences are densely connected. The main advantages of DBSCAN are:

Figure 7. DBSCAN simple.

1) You can cluster dense datasets of any shape. In contrast, clustering algorithms such as K-means are generally only suitable for convex datasets.

2) Anomaly points can be found during the clustering process, and they are not sensitive to abnormal points in the dataset.

3) There is no bias in the clustering results. In contrast, the initial value of clustering algorithms, such as K-means, has a significant influence on the clustering results.

DBSCAN's main disadvantages are:

1) If the density of the sample set is not uniform and the clustering distances are very different, the clustering quality is poor. At this time, DBSCAN clustering is generally not suitable.

2) If the sample set is large, the cluster convergence time is longer. At this time, the size of the KD tree or the ball tree established when searching the nearest neighbor can be improved by limiting the size.

3) Tuning parameters is slightly more complicated than traditional clustering algorithms, such as K-means. It mainly needs to adjust the distance threshold ϵ and the neighborhood sample number threshold MinPts. Different parameter combinations have a more significant effect on the final clustering.

3. Method and Approach

3.1 Dataset

For the insider threat detection, there are not many suitable datasets. So, we have used the insider threat dataset published by CERT (Carnegie Mellon University) for our research [14]. We used the "R4.2.tar.bz" dataset for our analysis. This dataset consists of six types of data records: HTTP, logon, device, file, email, and psychometric. This dataset contains all the activi-

ties of 1,000 employees within 17 months. The HTTP records contain user, PC, URL, and content from the website with time stamps. “Logon.csv” consists of the user’s “Logon” and “Logoff” activities with their PC with time stamps. “Logon” and “Logoff” are the two kinds of activities that can be found in the data records. “Logon” activity corresponds to the event of each user’s login or an event of one user unlock their screen while the “Logoff” event corresponds to the user’s logoff event. But, screen locks do not appear in this dataset. The third part of the dataset “device.csv” is a collection of data records of removable media device usage. It indicates connection and disconnection actions with the relevant user, PC, and time stamps. “file.csv” contains the details of file copies with date, user, PC, filename, and content. For figure out the relationship between the employee, the CERT dataset provides email communication records including from, to, cc and bcc fields. In the last part of the dataset, “psychometric.csv” provides psychometric scores based on the Big 5 personality traits for the definition of pers[onality]. Anomaly detection has been the main focus of many researchers’ due to its potential in detecting novel attacks. However, its adoption to real-world applications has been hampered due to system complexity as these systems require a substantial amount of testing, evaluation, and tuning prior to deployment. Running these systems over real labeled network traces with a comprehensive and extensive set of intrusions and abnormal behavior is the most idealistic methodology for testing in the 2013 IEEE Security and Privacy Workshops, a new dataset for internal intrusion detection was proposed in a paper published by researchers at Carnegie Mellon University [20]. The dataset has a complete time series and labels, and defines the scenarios for the red team’s abnormal behavior.

1. Users who don’t use mobile devices very often and get used to work normally suddenly start using mobile devices and start uploading data to WikiLeaks and leave in a short time.
2. Users start frequent visits to job search sites, and when they leave, they start to use mobile devices frequently (more frequently than usual) to transfer data.
3. The system administrator uses the mobile device to copy the keylogger to his supervisor’s computer due to negative emotions for the company or other reasons. The next day, through the collected information, log in as a supervisor and leave the company after sending a large number of panicked emails within the company.

3.2 Rule-based Feature Extraction

Eldardiry et al. [15] proposed a feature extraction method. Too many features may make the user probability unable to distinguish between normal behavior and abnormal behavior with insufficient data in [16]. This feature extraction method only takes the maximum and average values of some values and ignores the characteristics of the time series of the dataset. The right approach is to divide the behavior with as few features as possible clearly. Our feature extraction method is shown in Table 1.

For example, if the user logs in to the computer during the period from 8 to 9 o’clock, the user’s Feature is marked as 0. Because it is normal working hours, its Hidden Feature is 0. If the total activity of this user from 8 to 9 is N events, we can get data for N groups (Feature, Hidden Feature), and we calculate the proportion of each Feature in N events. Then, the probability of occurrence of event 0 is N_0/N . If the probability is 0, we will use the following methods to fill in the probability; if there are 16 feature classes with features 0-15, and the feature 4’s probability is 0, then the probability of feature 4 is $1/(N + 16)$. Similarly, the probability of feature 5 that is not 0 is calculated as $(N_5 + 16)/(N + 16)$. We extracted his probability data from 8 o’clock to 9 o’clock every day for more than one year, as shown in Figure 8, and used it for further cluster analysis.

3.3 Proposed Modeling Method

In view of the characteristics of the double-layer HMM, our modeling process is as follows:

1. The five parts of the dataset are classified according to users and arranged in chronological order.
2. According to the feature map, we use rules to convert features to numbers from 0 to 15.
3. Calculate the B parameters required by HMM. If the probability is 0, then Laplace’s theorem is used for processing.
4. Cluster analysis was performed after adding hidden state work on and work off to the B parameter.
5. Model the calculated parameters A, B, π .
6. Test the model with attack data and get the results.

The flowchart of our system is shown in Figure 9.

Table 1. Feature map

Feature number	Hidden Feature number	Feature name	Description
0	Work on: 0	Logon (self pc)	User logon user's own computer
1		Logoff (self pc)	User logoff user's own computer
2		Logon (other pc)	User logon other's own computer
3		Logoff (other pc)	User logoff other's own computer
4		Connect (self pc)	Device connected to user's own computer
5		Disconnect (self pc)	Device disconnected to user's own computer
6		Connect (other pc)	Device connected to other user's own computer
7		Disconnect (other pc)	Device disconnected to other user's own computer
8	Work off: 1	Email_In	Send email in company
9		Email_Out	Send email in outside
10		Normal-website	Device disconnected to user's own computer
11		Danger-website	Websites with a tendency to leak or provide high-risk software downloads
12		Job-website	Website for posting job information
13		File_self	User modifies files on their own computer
14		File_other	User modifies files on their other computer
15		Empty	Use 15 instead when the number of actions in a fixed time is insufficient

Figure 8. Probability dataset for all events at the same time each day by a single user.

4. Experiments

Through the process described in Section 3 of the article, we conducted experiments. The experimental environment used

Figure 9. The flowchart of proposed model.

Intel's 6-core CPU and 16 GB of memory, and the operating system used Ubuntu 18.04. Use Python as the language to write programs, and the IDE uses pycharm. The biggest advantage of Python is that it has a rich interface for scientific settlement. The event statistics of the dataset are shown in Table 2.

There are a total of 50 sets of attack data scenarios in the dataset. We extract the features in the form of a feature map and use it to identify each user after modeling.

The dataset contains a total of 1,000 users, and 50 of them have had abnormal behaviors for nearly one and a half years. Our model is to create a model for everyone to distinguish between users and other normal users, and between users and abnormal users. The experimental method is to test each other in a loop, so in order to show the results, we first find one of the representative users to explain. Our experimental goal is only one point, so that each user's model is sufficiently stable to distinguish between its own behavior, abnormal behavior, and normal behavior mixed with abnormal behavior.

First, we select the user HSB0196 (user'id), group the dataset, and calculate the hourly behavioral data model of this user from January 4, 2010 to May 17, 2011 based on hourly intervals. In

Table 2. Data set event statistics

Part	the number of event
Logon	854,860
File	445,582
Email	2,629,980
Device	405,381
HTTP	28,434,423

Table 3. The Result of DBSCAN

Class	Number (%)
0	86 (25)
1	125 (37)
2	121 (36)
3	6 (2)

Figure 10. Visualization of clustering results.

Figure 11. ROC curve.

other words, each user has 24 double-layer HMMs. As shown in Figure 8, all time points are integrated at 8 o'clock. The results after clustering using the DBSCAN algorithm are shown in Table 3. The epsilon is 0.9 and minPoints is 6.

This user's 8 o'clock has a total of 346 times. There are 8 data that cannot be classified. See Figure 10 for details.

According to the ROC curve shown in Figure 11, the results are highly credible.

We will use 50 behavioral data to attack users, 50 normal data, and 50 normal data mixed with abnormal data for testing. All of the 1,000 users, except the users with severe data loss, showed the effect shown in Figure 12. The upper part of the way is normal data, the middle is normal data mixed with abnormal

Figure 12. Experiment results.

Table 4. Result of all users ($n = 1,000$)

	number	accuracy
Normal user	950	94%
Abnormal user	50	99%

data, and the following is the probability fluctuation of abnormal data. After testing, the detection accuracy of all users is shown in Table 4.

5. Conclusions

This paper provides a new double-layer HMM structure to model user behavior. Our method belongs to anomaly detection. Compared with the Markov chain, our model is more efficient regarding probability. The experimental results show that when the amount of data becomes more significant, the result is more accurate. Compared with other papers, the algorithm is more accurate in distinguishing single users from other users. But a single clustering algorithm may not satisfy a wide variety of users. But the system has some disadvantages; when we face the malicious behavior of users without any data accumulation, we can do nothing about the attack. So, we need more experimental data to confirm our assumptions. In the future, we should consider using adaptive algorithms to model all users more accurately and solve the problem of new users or data-less users.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2018R1C1B3008718).

References

- [1] T. Seals, "75% of companies are insider threat victims," 2015, Available <https://www.infosecurity-magazine.com/news/75-of-companies-are-insider-threat/>
- [2] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 9-42, 2001. <https://doi.org/10.1142/S0218001401000836>
- [3] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov Models for Speech Recognition*. Edinburgh, UK: Edinburgh University Press, 1990.
- [4] Y. Wen, "Text mining using HMM and PMM," Ph.D. dissertation, University of Waikato, Hamilton, New Zealand, 2001.
- [5] H. Bunke and T. Caelli, *Hidden Markov Models: Applications in Computer Vision*. Singapore: World Scientific, 2001.
- [6] R. J. Boys, D. A. Henderson, and D. J. Wilkinson, "Detecting homogeneous segments in DNA sequences by using hidden Markov models," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 49, no. 2, pp. 269-285, 2000. <https://doi.org/10.1111/1467-9876.00191>
- [7] Z. Anming and J. Chunfu, "Study on the applications of hidden Markov models to computer intrusion detection," in *Proceedings of the 5th World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, Hangzhou, China, 2004, pp. 4352-4356. <https://doi.org/10.1109/WCICA.2004.1342335>
- [8] S. Shin, S. Lee, H. Kim, and S. Kim, "Advanced probabilistic approach for network intrusion forecasting and

- detection,” *Expert Systems with Applications*, vol. 40, no. 1, pp. 315-322, 2013. <https://doi.org/10.1016/j.eswa.2012.07.057>
- [9] M. H. Kabir, M. R. Hoque, K. Thapa, and S. H. Yang, “Two-layer hidden Markov model for human activity recognition in home environments,” *International Journal of Distributed Sensor Networks*, 2016. <https://doi.org/10.1155/2016/4560365>
- [10] E. Parzen, *Stochastic Processes*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.
- [11] S. Karlin and H. E. Taylor, *A First Course in Stochastic Processes*. St. Louis, MO: Elsevier, 2014.
- [12] J. Lamperti, “Markov transition function,” in *Stochastic Processes: A Survey of the Mathematical Theory*. New York, NY: Springer Science & Business Media, 2012, pp. 106-131.
- [13] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554-1563, 1966.
- [14] CERT Insider Threat Center, “The CERT Insider Threat Database,” 2011, Available <https://insights.sei.cmu.edu/insider-threat/2011/08/the-cert-insider-threat-database.html>
- [15] H. Eldardiry, E. Bart, J. Liu, J. Hanley, B. Price, and O. Brdiczka, “Multi-domain information fusion for insider threat detection,” in *Proceedings of 2013 IEEE Security and Privacy Workshops*, San Francisco, CA, 2013, pp. 45-51. <https://doi.org/10.1109/SPW.2013.14>
- [16] A. Gamachchi, L. Sun, and S. Boztas, “Graph based framework for malicious insider threat detection,” in *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*, Hilton Waikoloa Village, HI, 2017.
- [17] G. D. Forney, “The Viterbi algorithm: a personal history,” 2005, Available <https://arxiv.org/abs/cs/0504020>
- [18] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Harlow, UK: Pearson, 2014.
- [19] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, Portland, OR, 1996, pp. 226-231.
- [20] J. Glasser and B. Lindauer, “Bridging the gap: a pragmatic approach to generating insider threat data,” in *Proceedings of 2013 IEEE Security and Privacy Workshops*, San Francisco, CA, 2013, pp. 98-104. <https://doi.org/10.1109/SPW.2013.37>

Xiaoyun Ye was born in Yantai, China, in 1988. He received the master's degree in Computer Science from Gachon University, pursuing a PhD at Gachon University. His research interests include machine learning, cyber security, intelligent security network, data mining, and pattern

recognition.

E-mail: yxysun@gmail.com

Sung-Sam Hong was born in Seoul, Korea, in 1983. He received the master's degree in Computer Science from Gachon University, Korea in 2011 and the Ph.D. degree in Computer Science from Gachon University, Korea in 2016. He is a researcher professor in the Department of

Computer Engineering in Gachon University, Korea. His research interests include multimedia security, intelligent information security, machine learning, cryptology, data mining, and big data.

E-mail: sungsamhong0@gachon.ac.kr

Myung-Mook Han received M.S. degree in computer science from New York Institute of Technology in 1987 and Ph.D. degree in information engineering from Osaka City University in 1997, respectively. From 2004 to 2005, he was a visiting professor at Georgia Tech Information Security Center (GTISC), Georgia Institute of Technology.

Currently, He is a professor in the Department of Software, Gachon University, Korea. His research interests include information security, intelligent system, data mining, and big data.

E-mail: mmhan@gachon.ac.kr