applied sciences

MDPI

Article

A Novel Method for Detecting Advanced Persistent Threat Attack Based on Belief Rule Base

Guozhu Wang [1], Yiwen Cui [2], Jie Wang [2], Lihua Wu [1] and Guanyu Hu [2,*]

[1] School of Information Science Technology, Hainan Normal University, Haikou 571158, China; wangguozhu192002@hainnu.edu.cn (G.W.); lihuawu63@163.com (L.W.)
[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China; cui_yiwen2021@163.com (Y.C.); wangj1009@163.com (J.W.)
* Correspondence: huguanyu@guet.edu.cn

Abstract: Advanced persistent threat (APT) is a special attack method, which is usually initiated by hacker groups to steal data or destroy systems for large enterprises and even countries. APT has a long-term and multi-stage characteristic, which makes it difficult for traditional detection methods to effectively identify. To detect APT attacks requires solving some problems: how to deal with various uncertain information during APT attack detection, how to fully train the APT detection model with small attack samples, and how to obtain the interpretable detection results for subsequent APT attack forensics. Traditional detection methods cannot effectively utilize multiple uncertain information with small samples. Meanwhile, most detection models are black box and lack a transparent calculation process, which makes it impossible for managers to analyze the reliability and evidence of the results. To solve these problems, a novel detection method based on belief rule base (BRB) is proposed in this paper, where expert knowledge and small samples are both utilized to obtain interpretable detection results. A case study with numerical simulation is established to prove the effectiveness and practicality of the proposed method.

Keywords: advanced persistent threat; belief rule base; attack detection; network security

check for updates

Citation: Wang, G.; Cui, Y.; Wang, J.; Wu, L.; Hu, G. A Novel Method for Detecting Advanced Persistent Threat Attack Based on Belief Rule Base. *Appl. Sci.* **2021**, *11*, 9899. https://doi.org/10.3390/app11219899

Academic Editor: Frede Blaabjerg

Received: 13 August 2021
Accepted: 6 October 2021
Published: 22 October 2021

1. Introduction

The advanced persistent threat (APT) attack is receiving attention from security professionals, as attackers have shifted their targets from institutions with military backgrounds to businesses that focus on education, energy, computers, finance, and diplomacy. APT can be defined as an entity that engages in malicious, organized, and highly sophisticated long-term intrusion or exploitation operation to obtain information from a target organization [1]. There are two purposes of APT attacks, one is to steal critical data and the other is to destroy system infrastructure. By analyzing some very well-known APT attacks, such as TitanRain, Hydraq, Stuxnet, and the RSA SecureID Attack, the most typical common characteristics can be summarized as long term and multi stage [2], which means that an APT attack has strong concealment. Only when attackers are exposed or complete their missions may the intrusion be discovered [2,3]. On the other hand, it is also an unavoidable problem that APT attackers will use the latest vulnerabilities and tools. They are very difficult to detect [4]. Therefore, APT attack detection and defense research are very important for key institutions and network systems [5].

Currently, there are many methods attempting to solve the problem of APT detection.

(1) Qualitative analysis-based methods, which utilize expert knowledge for analyzing attack, study the relationship between network feature and attacks. For example, a hybrid strategy game-based model is proposed by [6] to compute the Nash equilibrium of network and possibility of APT attacks. It also proposed a data-fusion method to improve accuracy of the model. The literature [7] uses cumulative prospect theory to model APT attacks and proposes a policy hill-climbing-based detection scheme,

which improves the APT detection performance in dynamic games. The literature [8] utilizes four parallel classifiers to generate the events, then utilizes the correlation rules developed by experts to process the events, and finally, obtains the result by voting. Four classifiers are genetic programming, classification, and regression trees, support vector machine (SVM), and dynamic Bayesian game model. The literature [9] proposes a semantic-based correlation approach for detecting hybrid and low-level APT. They claim that this method is a big step towards the automatic detection of APT, while they mention that they did not take slow APT into account and could definitely not detect them. The literature [10] presents an APT detection system based on audit log data, which uses correlation information and kill-chain to process information and eventually output a high-level attack scenario graph for security experts to analyze. Qualitative analysis-based methods rely heavily on expert knowledge and intelligence, which are both strengths and weaknesses. Expert knowledge and intelligence can be utilized to detect unknown APT attack and reduce detection time. However, qualitative analysis-based methods are not accurate enough, which cannot detect APT accurately.

(2) Data-driven methods, which utilize data for training nonlinear model. The literature [11] proposes an intrusion detection system based on the decision tree by using analysis of behavior information to detect APT attacks. The literature [12] uses the continuous association rule mining algorithm to process intrusion detection events as attack scenarios and, then, predicts APT attacks by rules. The literature [13] proposes an innovative APT attack detection model based on a semi-supervised learning approach and complex networks characteristics, which is based on the shared nearest neighbor clustering algorithm. The literature [14] uses principal component analysis to reduce the dimension of data, then compares the performance of four methods, including SVM, naive Bayes, decision tree, and multi-layer perceptron (MLP). The literature [15] proposes a Bayesian network based weighted attack paths modeling technique to model APT attack paths. The literature [16] proposes a novel machine learning-based system to detect and predict APT attacks in a holistic approach, which uses a filter, cluster, and index (FCI) correlation framework to find the alerts that relate APT attack scenario, and finally, uses decision tree, SVM, k-nearest neighbors, and ensemble learning to detect APT. The literature [17,18] propose an APT detection frameworks by using unsupervised clustering and deep learning method with DNS logs, which develops a new feature that can represent the relationship between DNS request and response message. The literature [19] proposes a spatio-temporal association analysis to detect the APT attack in industrial network, which utilizes a frequent pattern (FP) growth algorithm to mine association rules and, finally, uses SVM for classification detection. The literature [20] uses feature extraction and normalization to process network traffic and, then, scores the hosts to obtain the final list of hosts receiving APT attack threats. The literature [21] presents an APT detection method based on the gradient boosting decision tree that uses temporal correlation features, traffic-based features, and combined features. Data-driven methods require a large number of samples to train the parameters of the detection model, which need sufficient data to obtain the optimal parameters to detect APT attack accurately. However, APT samples are hard to obtain due to their special characteristics mentioned above.

(3) Semi-quantitative analysis-based methods, which can both utilize expert knowledge and quantitative data. The literature [22] reconstructs APT attack scenarios using the alert correlation framework and, then, utilizes the hidden Markov model (HMM) to find the most consistent APT attack sequences and to predict the next phase of attacks. Traditional semi-quantitative analysis-based methods can combine expert knowledge and quantitative information to obtain more interpretable and accurate results with fewer samples, but it cannot exploit multiple uncertain information simultaneously. In addition, how to better embed expert knowledge into the model is also an urgent problem.

Therefore, to solve the above problems, a novel detection method based on belief rule base (BRB) is proposed in this paper, where a set of decision rules is firstly defined by mechanism analysis and expert knowledge, and then, a small number of quantitative data samples are used to optimize the model. Through a combination of expert knowledge and quantitative data, the proposed method can effectively solve the concealment problem caused by long-time and multi-stage characteristics of APT attack. Moreover, the proposed method also has the advantages of dealing with many kinds of uncertain information and the transparent reasoning process, which makes it possible to obtain more objective and accurate detection results with interpretability.

The remainder of this paper is organized as follows. In Section 2, basic problems about APT attack are described. In Section 3, the BRB model is introduced, and the BRB-based APT detection model is proposed. A simulation-based case study is designed to test the effectiveness of the proposed method in Section 4. Finally, Section 5 presents conclusions and directions for further research.

2. Problem Formulation

In a word, the methods described in Section 1 are not effective enough to detect long-time multi-stage APT attacks. There are still three challenges:

- **Small samples problem**, which makes parameter training of the model difficult to achieve the expected results. APTs reduce their chances of being detected by long and slow attack. Therefore, a defense system can only obtain a very small number of APT samples for a long time. The number of samples is usually not enough for training a data-driven model to achieve high detection accuracy.
- **Combining multiple information**, which is difficult for most existing methods. A variety of information exists in the process of APT attacks, such as expert experiences, domain knowledge, and quantitative data. In addition, there are various types of uncertainties in the exploitable information, such as the randomness of the moment of APT attack phase, the fuzziness between APT attacks and normal attacks, and the ignorance generated by zero-day attacks.
- **Interpretability of results**. If the model has no interpretability and cannot trace the reasons and process, the detection results will not be credible enough. A low-credibility model will lead to an increase in the false positive rate and false negative rate. On the other hand, the results of non-interpretation will affect the subsequent forensics of the APT attack.

To solve the above challenges, a novel method for detecting an APT attack based on belief rule base is proposed in this paper. The BRB model is a new intelligent expert system with characteristics of both an expert system and data-driven model, which can be used to solve the decision problems of complex system [23–26]. The proposed BRB-based APT detection model has the following features compared to traditional methods mentioned above:

- The BRB-based APT detection model has an advantage in solving small sample modeling problem due to embedding of expert knowledge. Using expert knowledge can set reasonable initial parameters, and only a small number of samples are needed to complete the model training, which can address the small sample problem.
- The BRB-based APT detection model is a semi-quantitative method that can handle a wide range of uncertain information by using expert knowledge and quantitative data. Uncertain information contains more useful features of APT attack and can express more objective results.
- The BRB-based APT detection model is a white box model that has good interpretability of result. The rules of BRB are designed in accordance with human knowledge, which can help experts to make targeted defense strategies. The reasoning process of BRB is transparent. Any rules that have been activated are visible. Experts can trace how the APT attack occurred and how the results were derived.

3. The BRB-Based APT Detection Model

3.1. Basic Principles of BRB

The BRB model consists of a number of rules established by experts. The decision conditions of rules consist of several antecedent attributes chosen by experts, and each attribute has some reference values. The consequents of rules are assigned to some belief degrees. It can be described as follows:

$$\begin{aligned}&R_k : \text{If } \left(x_1 \text{ is } A_1^k\right) \wedge \left(x_2 \text{ is } A_2^k\right) \wedge, \cdots, \wedge \left(x_M \text{ is } A_M^k\right)\\ &\text{Then } \left\{(D_1, \beta_{1,k}), (D_2, \beta_{2,k}), \cdots, (D_N, \beta_{N,k})\right\}\\ &\text{with rule weight } \theta_k (k = 1, 2, \cdots, L) \text{ and attribute weight } \delta_i (i = 1, 2, \cdots, M)\end{aligned} \tag{1}$$

where R_k denotes the kth rule of the belief rule base. $x_i (i = 1, \cdots, M)$ denotes the input attributes, and $A_i^k (i = 1, \cdots, M)$ denotes the reference value of attributes in the kth rule, where M denotes the number of attributes. $D_j (j = 1, \cdots, M)$ denotes the output of belief rule base, where N denotes the number of results. $\beta_{j,k}$ denotes the belief degree corresponding to each D_j.

When one input activates multiple rules, there will be several different results at the same time. Hence, simply using rules to represent expert knowledge is not enough, the results of every activated rule is used to reason the final result by the evidential reasoning (ER) rule. The ER rule is a multi-source information fusion method based on the theory of evidence and has been successfully applied on multiple attribute decision making [27,28]. The reasoning process of BRB by using the analytical ER approach is shown as follows:

$$\hat{\beta}_j = \frac{\mu \times \left[\prod_{k=1}^{L} \left(\omega_k \beta_{j,k} + 1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right) - \prod_{k=1}^{L} \left(1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]}{1 - \mu \times \left[\prod_{k=1}^{L} (1 - \omega_k)\right]} \tag{2}$$

$$\mu = \left[\sum_{j=1}^{N} \prod_{k=1}^{L} \left(\omega_k \beta_{j,k} + 1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right) - (N-1) \prod_{k=1}^{L} \left(1 - \omega_k \sum_{i=1}^{N} \beta_{i,k}\right)\right]^{-1} \tag{3}$$

where L denotes the total number of rules, and ω_k is the activate weight of the kth rule. $\hat{\beta}_j (j = 1, \cdots, N)$ is the final belief degree. ω_k can be calculated by the following formula.

$$\omega_k = \frac{\theta_k \prod_{i=1}^{M} (\alpha_{k,i})^{\delta_i}}{\sum_{k=1}^{L} \theta_k \prod_{i=1}^{M} (\alpha_{k,i})^{\delta_i}} \tag{4}$$

where $\alpha_{k,i} (i = 1, \cdots, M)$ denotes the degree to which the reference value is matched by the ith attribute of the kth rule. For qualitative information, match degrees can be given directly using expert knowledge, and for quantitative information, match degrees can be calculated using membership functions or expert rules.

Expert knowledge can be used to set rule weights, attribute weights, and result belief degrees, but expert knowledge is sketchy and cannot set the accurate parameters enough. Therefore, the P-CMA-ES algorithm can be utilized to optimize the parameters of the BRB detection model [29–31].

3.2. APT Detection Method Based on BRB

3.2.1. APT Attack Feature Selection

First, the key features that can describe the APT attack are selected as the antecedent attributes of the BRB detection model.

(1) **Traced path.** Attackers focus on high-value objects such as databases, servers, and significant hosts, which is one of features of APT. High-value objects can be represented by their location in the network if it is connected to the network. When the location in the network is determined, the value of the corresponding object is also determined. For example, the database is very significant because it stores essential data, and it can be represented on the network as a client data server, a backup server, or a data center server in the intranet. High-value objects need to be closely monitored to prevent them from being destroyed or stolen by attackers, so it is important to identify the source of access. Direct access is easy to detect, so APT attacks are usually hidden in normal access traffic to avoid detection. However, when tracing the source of traffic, anomalous traffic and normal traffic have a different access source somewhere, shown as Figure 1. Then, the traced path can be used as one of the features to detect APT. The location represented in the traffic is the IP address, which is the data to be collected.

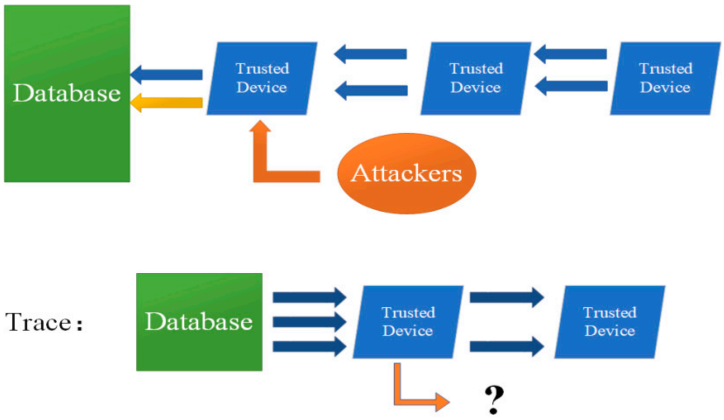

Figure 1. Traced path.

(2) **Traffic rate.** Just the traced path is not enough. The IP address can also be forged. Other features need to be used as a supplement to determine APT. The higher the rate of attack traffic in the normal traffic, the easier it is to detect. Since frequent anomalies can easily be assumed that an attack has occurred. The traffic from APT attacks usually represents only a small fraction of normal traffic to reduce the chances of detection. The rate of anomalous traffic to total traffic for a time period can be used as a feature to detect APT. When the rate is small, it means that the traffic is likely to be an APT attack. The rate can be calculated by counting traffic packets.

(3) **Average time interval.** APT attacks will last a long time as a way to reduce the correlation between attack traffic. APT attack traffic appears to be random, but it has a long average time interval in a long period of time. Therefore, the average time interval of traffic can be used as a feature to detect APT. The average time interval can be calculated by the time recorded when the traffic packet is captured.

(4) **Average transferred bytes.** The large number of transferred bytes should also be monitored. However, the alarm is not normally triggered when the number of bytes transferred is too small. APT can evade detection by transferring a small number of bytes at a time, splitting malicious code into many small pieces to avoid detection, and finally, reassembling it at the right time to achieve the purpose of attack, which is one of the common tactics of APT attackers. The average transferred bytes is one of features of APT, which can be calculated by traffic packets.

Based on above descriptions, four features are determined, including path, traffic rate, average time interval, and average transferred bytes, which can be extracted from the IP packets by experts. The reference values of attributes can be set, as shown in Table 1.

Table 1. Reference values.

Attributes	Reference Values		
Traced Path	Area-1	Area-2	...
Traffic Rate	Low	Medium	High
Average Time Interval	Low	Medium	High
Average Transferred Bytes	Low	Medium	High

The numerical value of reference value needs to be determined according to the actual network. "Low" corresponds to the smallest numeric value. "High" corresponds to the largest numeric value. "Medium" can be determined by expert knowledge.

3.2.2. Feature Extraction from IP Packets

The four features mentioned in the previous section can be extracted from the information obtained from IP packets. The traced path is derived from IP addresses, but IP addresses cannot be used directly as traced paths. It is difficult to construct rules on a host-by-host basis in a large network with many hosts because there will be too many attributes and reference values for BRB, which will lead to an exponential explosion problem. Host address requires information extraction. As mentioned above, the value of the host is important, and it can be represented on the network by an IP address. When there are many hosts of equal value, it is not necessary to determine the location of each host, but to treat them as an area. Converting IP address paths to area paths can solve the exponential explosion problem encountered in setting rules for BRB. Therefore, hosts are divided into different areas by expert knowledge based on their role and value. The IP address paths are then converted into the area paths by replacing the IP address with an area. The area paths are the traced paths. When the wrong sequence of areas appears, it can be considered as an anomaly. The traffic rate, average time interval, and average transferred bytes can be calculated from the information obtained from IP packets. The traffic over a period of time is a sample that contains both APT traffic and normal traffic. The processing flow can be represented as Figure 2.The detailed steps of feature extraction are as follows:

Step 1 Tracing anomalous paths by IP address. For example, a path is from A to B to C to D.
Step 2 Calculating the total bytes transferred by each path. The total number of bytes is the sum of the transferred bytes from A to B, B to C, and C to D.
Step 3 Calculating the time interval of every two neighboring nodes in the path and, then, sum up. The time interval is the time interval from A to B and B to C plus the time interval from B to C and C to D.
Step 4 Replacing malicious IP paths with malicious area paths.
Step 5 Fusing the same malicious area paths. Averaging over the transferred bytes and the time interval. Recording the number of fusions per path.
Step 6 Doing **steps 1** to **5** for normal traffic as well.
Step 7 Calculating the rate of paths among all paths using the number of fusions of paths.

By processing the original data, a dataset containing the four features proposed above is now obtained, which can be used to train the model.

3.2.3. Construction of Rules in BRB-Based APT Detection Model

The original belief rule base expert system based on the above four features can be established by expert knowledge. Depending on the features, the results can be divided into normal traffic, normal attacks, and APT attacks. APT is the long-duration and multi-stage attack. APT utilizes small transferred bytes, the low traffic rate, and long attack time to avoid the detection. Therefore, the sample in which the area path is abnormal, time interval is higher than the normal, the rate is lower than the normal, and the transferred bytes is low has a high probability of being an APT attack. The abnormal path means an unknown abnormal connection has appeared. It is likely that an attack has occurred. The higher time interval means abnormal connections are infrequent, which helps to hide them; however, normal connections are very frequent. The lower traffic rate and transferred bytes means anomalous connections make up only a small fraction of normal traffic when transferred, which reduces the chances of an intrusion detection system detecting it. Because of the large amount of traffic transferred, strict detection of each traffic by the intrusion detection system generates a large number of alarms, causing the normal transmission to not work properly. Then, the rules can be constructed such as follows:

$$\begin{aligned}&\mathrm{R}_1: \text{ IF } (areapath \text{ is } Abnormal) \wedge (trafficrate \text{ is } Low) \\ &\wedge (time\text{interval is } High) \wedge (transmitbytes \text{ is } Low) \\ &\text{Then Attack is}\{(No\_Attack,\ 0), (Normal\_Attack, 0), (APT, 1)\} \\ &\text{ with } \theta_1 = 1, \delta_1 = 1, \delta_2 = 1, \delta_3 = 1, \delta_4 = 1\end{aligned} \tag{5}$$

Those with normal attributes are normal samples, while those between normal and APT belong to normal attacks.

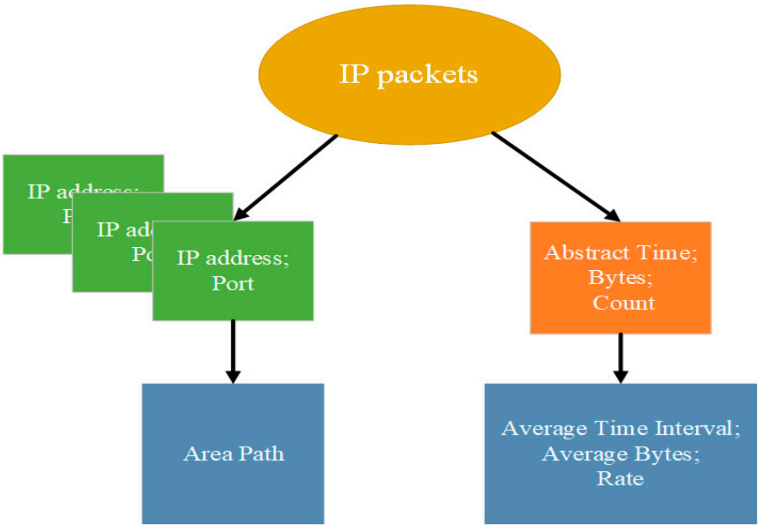

Figure 2. Features extraction.

The overall process is shown as Figure 3. Expert knowledge is significant, but not precise enough as a numerical parameter. The optimization algorithm discussed in Section 3.1 is used to optimize the parameters set by the expert to improve the accuracy of detection. The final trained model is used to detect APT attacks and gives interpretability results to trace back APT attacks.

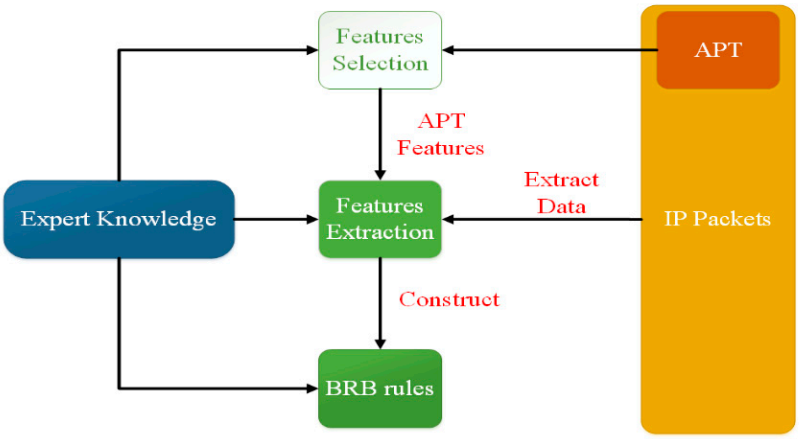

Figure 3. Process architecture.

4. Case Study

In this section, a dataset is used to prove the advantage of the proposed method. The raw dataset has 90 raw samples generated by Network Simulator 3. The processed sample set can be obtained after processing the original samples. The dataset has three kinds of samples, as shown in Table 2.

Table 2. Samples distribution.

Samples	No Attack	Normal Attack	APT
Amount	30	30	30

Network structure of the case is shown in Figure 4. The network includes two web servers, one SQL server, and five host groups. Web server stores resources for public network access. SQL server is used to store important internal information and update web server resources. The five host groups are management, technology, finance, human resources, and public relations, and their relationships are shown in Figure 5. They are divided into different areas depending on their importance. Management, finance, technology, and SQLserver belong to the important area. Human resources and public relations are in the secondary areas. Web server and public network are divided into a separate area.

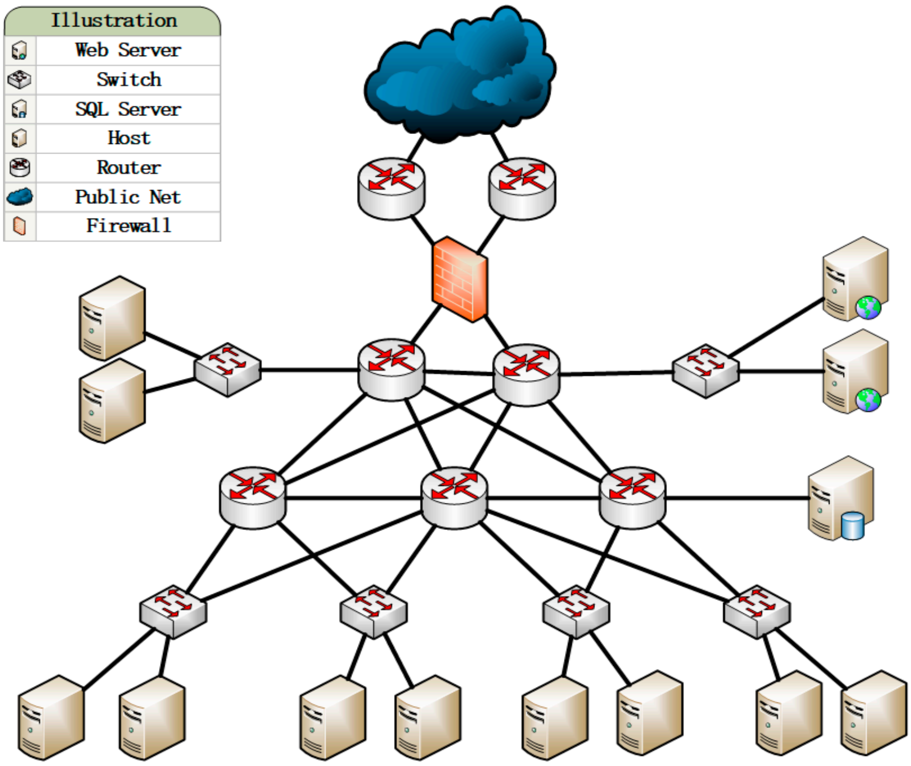

Figure 4. Network structure.

Both the server and the host are considered as nodes in the simulation software. The network connection between them is determined by deterministic codes. When the simulation software runs, the code generates network connections and sends data packets. One raw sample is the network traffic in the network in a fixed period of time, which includes many flows. The no-attack raw sample has three traffic counts, 42, 84, and 126. The other two samples are the same. Network connections are divided into normal connections and attack connections. Network connections contain three settable attributes: address, bandwidth, and duration. The address and bandwidth of a normal connection are deterministic. The address of an attack connection is artificially randomly selected.

Some possible APT paths are shown as red lines in Figure 6. The bandwidth of an attack connection is deterministic. The duration of a network connection is the base time plus an offset time. The base time of a normal connection is deterministic. The base time of an attack connection is artificially randomly chosen. The offset time is a pseudo-random number selection within an artificially specified random range. The settings of all the above parameters are subject to expert knowledge.

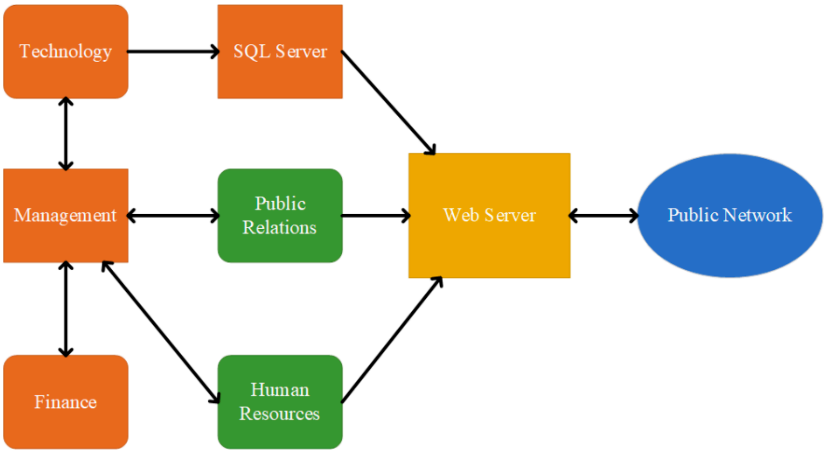

Figure 5. Relationships of units.

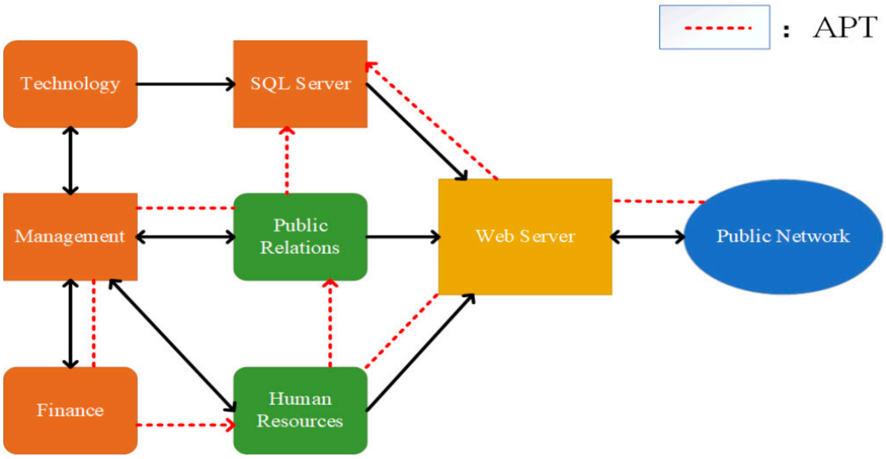

Figure 6. Possible APT paths.

Normal attacks and APT attacks are designed according to their respective characteristics. APT attackers are more patient and careful than normal attackers. APT has a higher focus on important resources. APT has a longer attack cycle and less traffic than normal attacks. In this case, the path of the APT attack tends to be more towards the most important areas, with a larger time interval, lower rate, and less traffic. Normal traffic has the most comprehensive path, with smaller time intervals, higher rate, and larger traffic. Normal attacks fall somewhere in between. The attack traffic generation code is inserted into the normal code according to the above knowledge.

The 90 raw data samples are then generated via NS3. This is acceptable, although there are many deterministic elements. First, there are deterministic standards for bandwidth and addresses in real networks. In this example, just a few values are chosen as criteria. Second, the network and normal business processes are designed by experts. Therefore, it is reasonable to design the normal network connection based on expert knowledge in this example. Third, APT attacks are launched by humans and are influenced by the attacker's expert knowledge. Therefore, it is acceptable to determine the attack connection based on expert knowledge. Finally, the random offset time complements the network fluctuations that exist in practice. Additionally, it increases the randomness of launching APT attacks in this example.

The raw data need to be processed, as discussed in Section 3.2. Combining information of two IP packets to obtain the path, time interval, and total number of bytes within each raw sample. For examples, packet_1 is from A to B and packet_2 is from B to C. Then, the path is from A to B to C. The time interval calculates by $\text{Abs\_time}_\text{B} - \text{Abs\_time}_\text{A} - \text{Duration}_\text{A}$ (which Abs_time is the time of packets start sending). The byte totals are directly summed. The IP address is, then, replaced by the area. The same area paths are fused to calculate the average time interval, average total number of bytes, and rate of a path, then selecting the data corresponding to one area path as a processed sample. The dataset used below consists of processed samples. The dataset includes area difference, rate difference, average time interval difference, and average total transferred bytes difference, as shown in Figures 7–10.

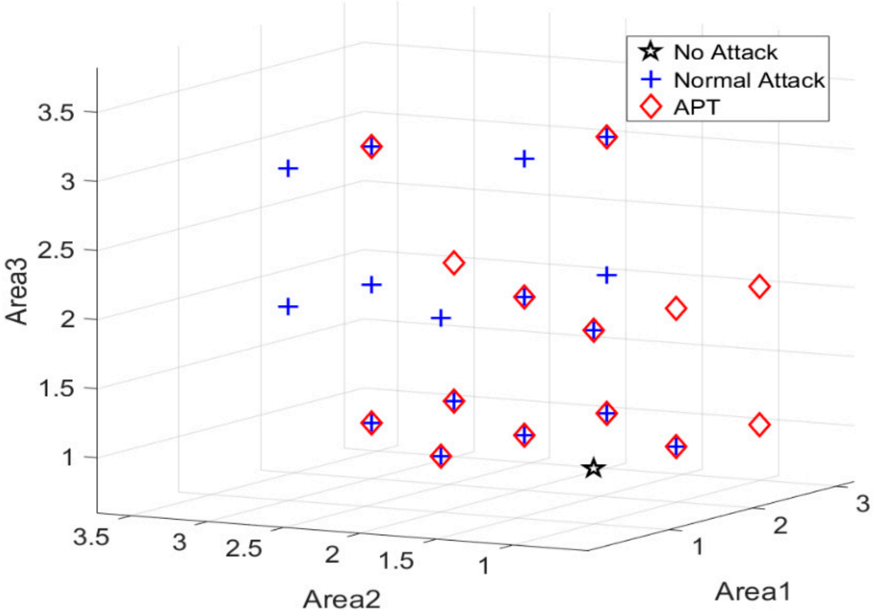

Figure 7. The area difference.

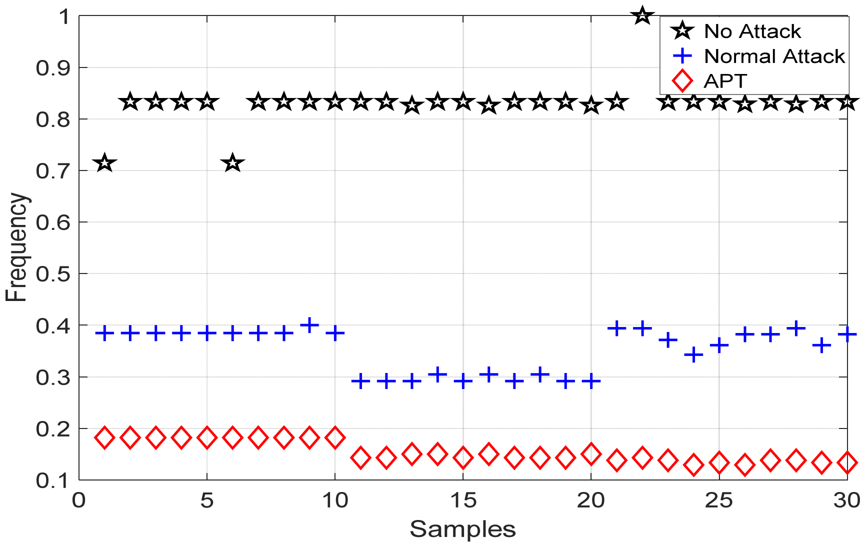

Figure 8. The rate difference.

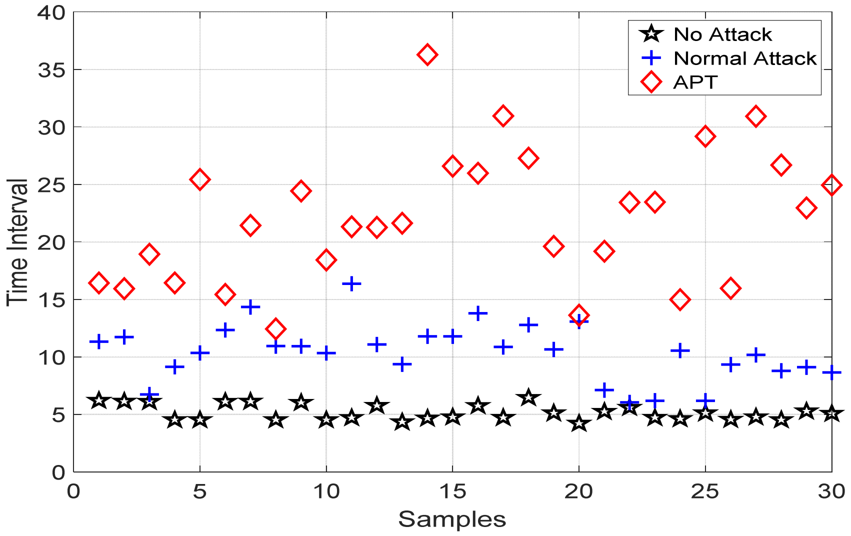

Figure 9. The average time interval difference.

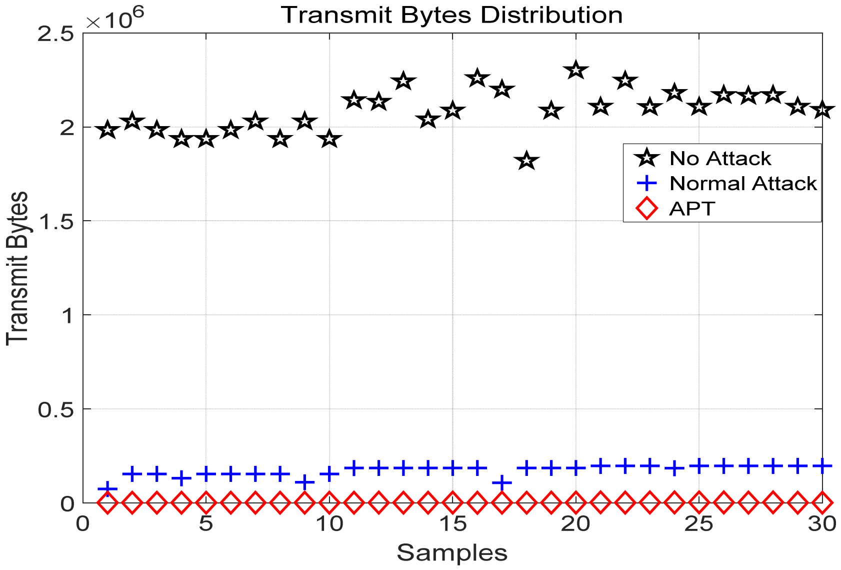

Figure 10. The average transferred bytes difference.

According to Figure 7, it can be seen that the regional path does not include all possibilities, but this is reasonable because there is expert knowledge as a supplement. The paths of APT and normal attacks are partially overlapped. It is known from expert knowledge that the normal path should have an overlapping part with the attack path, but Figure 7 does not contain it. Figure 8 shows that No Attack has the highest rate and APT is the lowest. The rate of normal attacks is centered. As shown in Figure 9, the time interval of APT is random, while the time interval of normal data is regular. The time interval of Normal Attack is somewhere in between. In Figure 10, the transferred bytes of APT is very small and the transferred bytes of No Attack is large. The transferred bytes of Normal Attack are slightly higher than those of an APT.

According to the expert knowledge derived from above analysis to establish the BRB detection model, the reference values are derived according to Figures 7–10 and are shown in Tables 3–6. Area-1, area-2, and area-3 have the same reference values in Table 3. The values of all samples fall within the reference value interval in Tables 4–6. The reference value of "Low" in Table 6 is lower than the lowest value of transferred bytes in all samples, which is set to 1024 bytes. The reference value of "High" is set to 2.5 × 1024 × 1024 bytes over the maximum value of transferred bytes in all samples. Additionally, the reference value of “Medium” is set to 200 × 1024 according to the value of Normal Attack in Figure 10 and expert knowledge. Semantic results are replaced with numbers to facilitate program processing as shown in Table 7. The reference values in Table 7 are used only as a distinction of the results and have no other meaning.

Table 3. Reference values of the area.

Linguistic Terms	Important Area	Secondary Area	Server	Public Network
Values	1	2	3	4

Table 4. Reference values of the rate difference.

Linguistic Terms	Low	Medium	High
Values	0.1	0.4	1

Table 5. Reference values of the time interval difference.

Linguistic Terms	Low	Medium	High
Values	0.1	10	38

Table 6. Reference values of the transferred bytes difference.

Linguistic Terms	Low	Medium	High
Values	1024	204,800	2,621,440

Table 7. Reference values of the attack.

Linguistic Terms	No Attack	Normal Attack	APT
Values	1	2	3

So, the rules can be described as follows:

$$\begin{aligned}
&R_k : \text{If } \left(area_1 \text{ is } A_1^k\right) \wedge \left(area_2 \text{ is } A_2^k\right) \wedge \left(area_3 \text{ is } A_3^k\right) \wedge \left(frequency \text{ is } A_4^k\right) \\
&\wedge\left(time\_interval \text{ is } A_5^k\right) \wedge \left(transmit\_bytes \text{ is } A_6^k\right) \\
&\text{Then Attack is } \left\{(No\_Attack, \beta_{1,k}), (Normal\_Attack, \beta_{2,k}), (APT, \beta_{3,k})\right\} \\
&\text{with rule weight } \theta_k (k = 1, 2, \cdots, L) \text{ and attribute weight } \delta_i (i = 1, 2, \cdots, M)
\end{aligned} \tag{6}$$

The BRB detection model has 1728 rules combined by the above six attributes. Expert knowledge is used to set initial weights and belief degrees for each rule. Some of rules are shown in Table 8. Expert knowledge only gives approximate values, which are imprecise. For rule 1 in Table 8, when the area-1, area-2, area-3, etc., are determined, the expert is uncertain about whether an apt attack actually occurred. However, it is certain that it is not an normal attack. So, the belief degree is set to 0.5, 0, and 0.5.

The data are randomly sorted as the input data. The top 25 of the input data are taken as the testing set and the remaining part as the training set. The P-CMA-ES algorithm is utilized to optimize the parameters to get the better results. The model is optimized for 80 rounds. Then, the 2nd to 26th samples of the input data are used as the testing set and the rest as the training set. It continues until, finally, the last 25 samples of the input data are used as the test set and the rest as the training set. A total of 65 rounds of training and testing are completed, and the average of the results is shown in Table 9. According to Table 9, the No Attack samples are all detected, but some other attacks are incorrectly detected as No Attack. The Normal Attack samples are detected in 77.88% of the total number of Normal Attack samples, but with 95.2% precision. The APT samples are detected in 93.84% of the total number of APT samples, with 87.25% precision.

Table 8. Initial parameters of BRB expert system.

Rule	Weight	Area1	Area2	Area3	Rate	Time Interval	Transferred Bytes	{No Attack, Normal Attack, APT}
1	1	1	2	1	High	Low	High	0.5, 0, 0.5
2	1	2	2	3	High	Medium	High	1, 0, 0
3	1	3	2	2	Low	High	High	0, 0.5, 0.5
4	1	1	1	1	Medium	Medium	Low	0.5, 0.5, 0
5	1	4	3	2	Low	Low	Medium	0, 0.5, 0.5
6	1	2	2	2	Medium	Low	High	1, 0, 0

Table 9. The evaluation of the results.

Items	No Attack	Normal Attack	APT
Precision	93.06%	95.2%	87.25%
Recall	100%	77.88%	93.84%
F1-score	96.03%	84.66%	89.87%

Compare experiment is necessary. Two models are selected to compare with the BRB which are SVM and MLP. They are typical machine learning algorithms. The parameters of these models are shown in Tables 10 and 11. "kernel" in Table 10 has better results when set to "rbf", which is the result of comparison with other functions. "C" and "gama" are used to prevent over-fitting. "decision_function_shape" is set to "ovr", indicating the division of a class from other classes at multiple classification. In Table 11, when the sample is not large, "solver" is set to "Lbfgs" with better effect. The values of "activation", "alpha", and "hidden_layer" are the better parameters under multiple tests. The training and testing sets are standardization and, then, fed to SVM and MLP for training and testing. Then, the comparisons of evaluations are shown in Tables 12–15.

Table 10. The parameters of SVM.

Items	C	Kernel	Gama	Decision_Function_Shape
Values	1	rbf	1/6	ovr

Table 11. The parameters of MLP.

Items	Activation	Solver	Alpha	Hidden_Layer
Values	tanh	Lbfgs	1e-5	11

Table 12. The comparison of precision rate.

Items	Precision		
	No Attack	Normal Attack	APT
BRB	93.06%	95.2%	87.25%
SVM	100%	90.94%	90.69%
MLP	100%	97.11%	91.02%

Table 13. The comparison of recall rate.

Items	**Recall**		
	No Attack	**Normal Attack**	**APT**
BRB	100%	77.88%	93.84%
SVM	100%	89.84%	90.56%
MLP	100%	89.93%	96.14%

Table 14. The comparison of F1-score.

Items	**F1-Score**		
	No Attack	**Normal Attack**	**APT**
BRB	96.03%	84.66%	89.87%
SVM	100%	88.91%	89.26%
MLP	100%	92.23%	92.45%

Table 15. The comparison of accuracy.

Models	**BRB**	**SVM**	**MLP**
Accuracy	91.14%	93.23%	95.32%

From the numerical value of the results, BRB does not dominate. In Table 12, the prediction precision of BRB for No Attack is only 93.06%, which is smaller than the 100% of SVM and MLP. In Table 13, the recall of No Attack is 100%. This indicates that BRB detected all correct No Attack samples, but also detected some other samples incorrectly as No Attack. In Table 12, BRB has a precision of 95.2% between SVM and MLP for Normal Attack and 87.25% for APT. In Table 13, BRB has a recall rate of only 77.88% for Normal Attack, and a detection accuracy of 93.84% for APT between SVM and MLP. F1-score is defined as the harmonic mean between precision and recall as shown in Table 14. BRB has an F1-score of 96.03% for No Attack, an F1-score of 84.66% for Normal Attack, and an F1-score of 89.87% for APT between SVM and MLP. Accuracy is the ratio of the number of correctly classified samples to the total number of samples as shown in Table 15. The accuracy of BRB is 91.14%, which is lower than the 93.23% of SVM and 95.32% of MLP. Although the accuracy of BRB is not the best, it is acceptable and the results are interpretable. The high accuracy obtained by SVM and MLP with very few samples is difficult to explain.

Multi-layer perceptron is initialized using random numbers. This approach implies that the starting point of the search results with suitable weights is random. The MLP model does not receive sufficient training with a limited sample size, so the final weights differ from the most suitable ones. However, in this case, the MLP achieved the highest accuracy with a small sample. This is difficult to interpret and does not provide valid suggestions for improving the defense plan. The SVM model with the "rbf" kernel has the same problem. In the case of small samples, there are no more samples to verify whether MLP and SVM really learn the appropriate parameters. MLP and SVM have high accuracy, but the reliability of the results is doubtful.

BRB is different from SVM and MLP and is built from expert knowledge. Therefore, the initial values of the BRB model have received a large amount of expert knowledge constraints. This approach already allows BRB to fit functions close to the appropriate weights when BRB is not yet trained. BRB can achieve high accuracy after training with a small number of samples. The accuracy of the BRB model is very dependent on expert knowledge. If the expert knowledge is biased, the results of the model are not good. In this example, the BRB model has 91.14% accuracy, which is lower than SVM and MLP. However, the BRB model is interpretable. Based on the training process, we believe that it is caused by the bias of the expert knowledge. The biased expert knowledge makes the scope of the constraints shifted. This bias still exists after optimization and has an impact on the final results. The results can help experts to validate and adjust their own knowledge, which is beneficial for updating BRB rules and correcting the defense plan.

The results illustrate that BRB obtained a reliable acceptable accuracy of 91% in the small sample case. MLP and SVM have high accuracy but cannot be interpreted and are unreliable. BRB uses expert knowledge to set the initial values of the parameters so that only a small number of samples are needed to complete the training, solving the small sample problem. BRB utilizes the membership function to deal with the randomness of the APT attack time. The SVM and MLP models are fit-based, while the BRB model is rule-based, and its interpretation is better than the fitted model. The BRB model is white box and has good interpretability, which facilitates the analysis and traceability of BRB results.

5. Conclusions

In this paper, a novel method for detecting APT attack based on the belief rule base model is proposed. The BRB-based APT detection model has the advantages of comprehensive utilization of information and generation of interpretable results. To overcome the difficulties of APT detection, we investigate the characteristics of APT attack in network traffic and construct the BRB model that can embed the expert knowledge and effectively express all kinds of uncertainty. Then, the BRB model is trained with small samples to obtain more accurate results. Two different algorithms are used to detect APT in a very small sample size to compare with the proposed method. The experiments show that the BRB-based method has a reliable acceptable accuracy. Further, the BRB-based APT detection model can help with computer forensics because the results have good interpretability and every rule can be regarded as an evidence. BRB combines expert knowledge with artificial intelligence, which is also a typical application of human–machine hybrid intelligence in the field of network security.

Our work provides a reasonable method for APT attack detection in traffic, which can obtain valid test results in the case of limited samples. However, there is still some work to be done next. First, there are many parameters of rules to be set by experts, which takes a lot of time and effort from experts. Automatic generation algorithms should be used in the parameter setting of rules. Second, expert knowledge bias needs to be taken into account when setting rules. The source of objective expert knowledge is an issue. Third, the impact of standardization of expert knowledge on the results is to be considered. Finally, APT contains many different types. In this paper, how to detect APT attacks in traffic are discussed. The next step is to detect APT in both traffic data and host data.

Author Contributions: Conceptualization, G.W. and G.H.; methodology, G.W. and Y.C.; software, G.W. and G.H.; validation, G.W., Y.C. and G.H.; formal analysis, Y.C.; investigation, G.W., Y.C. and J.W.; resources, G.H.; data curation, Y.C. and J.W.; writing—original draft preparation, G.W.; writing—review and editing, G.W.; visualization, Y.C. and J.W.; supervision, G.H. and L.W.; project administration, G.H. and L.W.; funding acquisition, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Hainan Province under Grant 2019CXTD405, China Postdoctoral Science Foundation under Grant 2020M673668, and Guangxi Key Laboratory of Trusted Software under Grant KX202050, and the key research and development plan of HaiNan under Grant ZDYF2019007.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used during this study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahmad, A.; Webb, J.; Desouza, K.C.; Boorman, J. Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack. *Comput. Secur.* **2019**, *86*, 402–418. [CrossRef]
2. Alshamrani, A.; Myneni, S.; Chowdhary, A.; Huang, D. A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1851–1877. [CrossRef]
3. Stojanović, B.; Hofer-Schmitz, K.; Kleb, U. APT datasets and attack modeling for automated detection methods: A review. *Comput. Secur.* **2020**, *92*, 101734. [CrossRef]
4. Chakkaravarthy, S.S.; Sangeetha, D.; Vaidehi, V. A Survey on malware analysis and mitigation techniques. *Comput. Sci. Rev.* **2019**, *32*, 1–23. [CrossRef]
5. Rubio, J.E.; Alcaraz, C.; Roman, R.; Lopez, J. Current cyber-defense trends in industrial control systems. *Comput. Secur.* **2019**, *87*, 101561. [CrossRef]
6. Lv, K.; Chen, Y.; Hu, C. Dynamic Defense Strategy against Advanced Persistent Threat Under Heterogeneous Networks. *Inf. Fusion* **2019**, *49*, 216–226. [CrossRef]
7. Xiao, L.; Xu, D.; Mandyam, N.B.; Poor, H.V. Attacker-Centric View of a Detection Game against Advanced Persistent Threats. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2512–2523. [CrossRef]
8. Sharma, P.K.; Moon, S.Y.; Moon, D.; Park, J.H. DFA-AD: A distributed framework architecture for the detection of advanced persistent threats. *Clust. Comput.* **2017**, *20*, 597–609. [CrossRef]
9. Lajevardi, A.M.; Amini, M. A semantic-based correlation approach for detecting hybrid and low-level APTs. *Future Gener. Comput. Syst.* **2019**, *96*, 64–88. [CrossRef]
10. Milajerdi, S.M.; Gjomemo, R.; Eshete, B.; Sekar, R.; Venkatakrishnan, V. HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 1137–1152. [CrossRef]
11. Moon, D.; Im, H.; Kim, I.; Park, J.H. DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *J. Supercomput.* **2017**, *73*, 2881–2895. [CrossRef]
12. Kim, Y.H.; Park, W.H. A study on cyber threat prediction based on intrusion detection event for APT attack detection. *Multimed. Tools Appl.* **2014**, *71*, 685–698. [CrossRef]
13. Zimba, A.; Chen, H.; Wang, Z.; Chishimba, M. Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics. *Future Gener. Comput. Syst.* **2020**, *106*, 501–517. [CrossRef]
14. Chu, W.L.; Lin, C.J.; Chang, K.N. Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine. *Appl. Sci.* **2019**, *9*, 4579. [CrossRef]
15. Zimba, A.; Chen, H.; Wang, Z. Bayesian network based weighted APT attack paths modeling in cloud computing. *Future Gener. Comput. Syst.* **2019**, *96*, 525–537. [CrossRef]
16. Ghafir, I.; Hammoudeh, M.; Prenosil, V.; Han, L.; Hegarty, R.; Rabie, K.; Navarro, F.J.A. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Gener. Comput. Syst.* **2018**, *89*, 349–359. [CrossRef]
17. Yan, G.; Li, Q.; Guo, D.; Li, B. AULD: Large Scale Suspicious DNS Activities Detection via Unsupervised Learning in Advanced Persistent Threats. *Sensors* **2019**, *19*, 3180. [CrossRef] [PubMed]
18. Yan, G.; Li, Q.; Guo, D.; Meng, X. Discovering Suspicious APT Behaviors by Analyzing DNS Activities. *Sensors* **2020**, *20*, 731. [CrossRef]
19. Wang, X.; Liu, Q.; Pan, Z.; Pang, G. APT attack detection algorithm based on spatio-temporal association analysis in industrial network. *J. Ambient. Intell. Humaniz. Comput.* **2020**. [CrossRef]
20. Marchetti, M.; Pierazzi, F.; Colajanni, M.; Guido, A. Analysis of high volumes of network traffic for advanced persistent threat detection. *Comput. Netw.* **2016**, *109*, 127–141. [CrossRef]
21. Lu, J.; Chen, K.; Zhuo, Z.; Zhang, X. A temporal correlation and traffic analysis approach for APT attacks detection. *Clust. Comput.* **2019**, *22*, 7347–7358. [CrossRef]
22. Ghafir, I.; Kyriakopoulos, K.G.; Lambotharan, S.; Aparicio-Navarro, F.J.; AsSadhan, B.; Binsalleeh, H.; Diab, D.M. Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats. *IEEE Access* **2019**, *7*, 99508–99520. [CrossRef]
23. Zhou, Z.-J.; Hu, G.-Y.; Hu, C.-H.; Wen, C.-L.; Chang, L.-L. A Survey of Belief Rule-Base Expert System. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4944–4958. [CrossRef]
24. Hu, G.Y.; Zhang, B.C.; Zhou, Z.J.; Wang, W.F.; Qiao, P.L. *Network Security Situational Awareness Based on Belief Rule Base*; The Science Publishing Company: Beijing, China, 2017. (In Chinese)
25. Zhou, Z.-J.; Hu, C.-H.; Zhang, B.-C.; Xu, D.-L.; Chen, Y.-W. Hidden behavior prediction of complex systems based on hybrid information. *IEEE Trans. Cybern.* **2013**, *43*, 402–411. [CrossRef]
26. Feng, Z.; Zhou, Z.J.; Hu, C.H.; Yin, X.; Hu, G.; Zhao, F. Fault Diagnosis Based on Belief Rule Base with Considering Attribute Correlation. *IEEE Access* **2018**, *6*, 2055–2067. [CrossRef]
27. Yang, J.B.; Singh, M.G. An evidential reasoning approach for multiple-attribute decision making with uncertainty. *IEEE Trans. Syst. Man Cybern.* **1994**, *24*, 1–18. [CrossRef]
28. Yang, J.B.; Xu, D.L. Evidential reasoning rule for evidence combination. *Artif. Intell.* **2013**, *205*, 1–29. [CrossRef]

29. Chang, L.-L.; Zhou, Z.-J.; Chen, Y.-W.; Liao, T.-J.; Hu, Y.; Yang, L.-H. Belief Rule Base Structure and Parameter Joint Optimization Under Disjunctive Assumption for Nonlinear Complex System Modeling. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1542–1554. [CrossRef]
30. Yang, J.-B.; Liu, J.; Xu, D.-L.; Wang, J.; Wang, H. Optimization Models for Training Belief-Rule-Based Systems. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2007**, *37*, 569–585. [CrossRef]
31. Hu, G.-Y.; Zhou, Z.-J.; Zhang, B.-C.; Yin, X.-J.; Gao, Z. A method for predicting the network security situation based on hidden BRB model and revised CMA-ES algorithm. *Appl. Soft Comput.* **2016**, *48*, 404–418. [CrossRef]