IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 August 2023, accepted 13 August 2023, date of publication 18 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306593

RESEARCH ARTICLE

BAN: Predicting APT Attack Based on Bayesian Network With MITRE ATT&CK Framework

YOUNGJOON KIM[1], INSUP LEE[1], (Student Member, IEEE), HYUK KWON[1], KYEONGSIK LEE[1], AND JIWON YOON[2]

[1]Cyber Technology Center, Agency for Defense Development, Seoul 05661, Republic of Korea
[2]School of Cybersecurity, Korea University, Seoul 02841, Republic of Korea

Corresponding authors: Kyeongsik Lee (n0fate@add.re.kr) and Jiwon Yoon (jiwon_yoon@korea.ac.kr)

This work was supported by the Agency for Defense Development, Republic of Korea.

ABSTRACT Since cyberattacks have become sophisticated in the form of advanced persistent threats (APTs), predicting and defending the APT attacks have drawn lots of attention. Although there have been related studies such as attack graphs, Hidden Markov Models, and Bayesian networks, they have four representative limitations; (i) non-standard attack modeling, (ii) lack of data-driven approaches, (iii) absence of real-world APT dataset, and (iv) high system dependability. In this paper, we propose Bayesian ATT&CK Network (BAN) which is based on system-independent data-driven approach. Specifically, BAN is based on Bayesian network, which adopts structure learning and parameter learning to model APT attackers with the MITRE ATT&CK® framework. The trained BAN aims to predict upcoming attack techniques and derives corresponding countermeasures. In addition, we prepare datasets via both automatic and manual labeling to overcome the data insufficiency issues of APT prediction. Experimental results show that BAN successfully contributes to handling APT attacks, given the best parameters extracted from extensive evaluations.

INDEX TERMS Attack prediction, advanced persistent threat, ATT&CK framework, Bayesian network, cyber threat intelligence.

I. INTRODUCTION

The global pandemic has accelerated the transition to digital and remote work, making cyberspace more critical than ever. After the pandemic, numerous people have been forced to work and live remotely. Moreover, various businesses have had to adapt by shifting their business online. As a result, much sensitive data have been moved to cyberspace, and reliance on cyberspace has increased dramatically. Unfortunately, as much information has transferred to cyberspace, cyber threats are also advancing.

Existing cyber threats have been mostly regarded as simple and single attacks. However, as cyber threats have become more sophisticated, targeted, and persistent, an Advanced Persistent Threat (APT) has recently become the most significant cyber threat. APT is an advanced and long-lasting cyber threat developed by a highly organized, sophisticated, and well-resourced group [1]. APT attacks have evolved as more sophisticated and frequent since their first appearance, causing considerable damage to countries and companies. In addition, APT attacks often occur for political and social motives rather than financial returns, resulting in various types of damage. The most representative example is the Democratic National Committee (DNC) hacking by Russian attackers just before the 2016 US presidential election [2].

Although APT attacks are growing daily, it is challenging to defend against them. Unlike conventional cyberattacks, APT attacks utilize almost all of the techniques used in existing cyberattacks. Furthermore, an APT attack can effectively bypass the attack detection system because they are often highly targeted and long-lasting. That is, the APT attackers can focus on the most vulnerable part of the system after collecting the target information. Therefore, it is tough for the defender to detect and respond to these APT attacks.

Due to the difficulty of detecting the APT attack, reactive attack detection and defense systems cannot effectively

The associate editor coordinating the review of this manuscript and approving it for publication was Wen Chen.

defend against APT attacks. We must abandon the false belief that real-time attack detection and defense are perfect. Instead, the defenders must predict the next attack based on the information and design a proactive defense method. By predicting the attacks, it is possible to cope with a failure in attack detection, minimize the damage of the attacks through a preemptive response, and finally prevent the attacker from achieving their ultimate goal.

Several studies have been conducted to predict cyberattacks. Most notably, many studies predict the next cyberattack based on the attack graph [3], [4], [5]. Also, some studies probabilistically infer the next attack based on machine learning, such as Hidden Markov Model [6], [7], [8] and Bayesian network [9], [10], [11]. Recently, attack prediction research is also being developed using deep learning [12], [13].

However, these cyberattack prediction methods have several limitations. First, they did not model the attacker's behavior in a general, standard framework. Some models can only predict the presence or absence of an attack [14], [15], or the predefined stage of an attack [6], [7], [8]. Also, several studies based on an Attack graph [3], [4] can model detailed attacks, but these studies manually model attacks by researchers' opinions. This lack of standardization can lead to inconsistent modeling methods depending on the individual expert, making it challenging to integrate with other defense systems.

Second, existing studies utilize a manual method in which experts subjectively determine the structures or parameters of the model rather than data-driven learning. For example, several studies based on attack graph [3], [11] generate an attack graph based on the security experts' knowledge, not the objective data. As a result, these studies require high-educated security experts to manually generate attack graphs based on their deep knowledge of the defense environment.

Third, existing studies do not utilize actual APT attack data that occurred in the real-world. Although some studies [6], [7] have utilized raw alert data (IDS logs), these data are not able to represent the actual behavior of the APT attackers. Additionally, researches that utilizes an attack graph [3], [11] to model the specific attack actions has not been validated through data. Accordingly, it is unknown whether the models will be effective in the real-world situations.

Finally, the prediction systems depend on the network structure and the detection system. For example, since the attack graph is dependent on the network and host environment, the studies based on attack graph [3], [4], [5] require a new model if the environment is different. Also, studies that predict cyberattacks based on raw alerts [6], [7] depend on a specific detection system of training datasets. If the detection system, such as IDS, has been changed, the model does not work correctly and has to be retrained. Therefore, it is unsuitable for the current companies' environment, where the enterprise network structures and detection systems are frequently changed.

To this end, we propose the Bayesian ATT&CK Network (BAN). which is a Bayesian network-based attack prediction model, to overcome these limitations. BAN models the attacker's behavior in detail based on a global standard framework, MITRE ATT&CK® [16]. In addition, we have constructed our own datasets that represent real APT cases. These datasets were created from reports analyzing previous APT attacks. Therefore, since the MITRE ATT&CK and our datasets do not depend on the network environment, our model can be used independently from network structures and detection systems. In addition, we maximized the objectivity and usability of the model by introducing fully automated data-driven learning. Also, BAN can predict not only the next attacks but also the attacker's ultimate goal. Finally, we also maximized the usefulness of BAN by offering appropriate defense techniques to the defender. Hence, we can operate the BAN in four aspects: predict the next attacks, the next defenses, the goal attacks, and the goal defenses. We performed experiments and evaluations using the prepared unbiased dataset to verify the performance of the proposed model.

The contributions of our study are as follows.

- We have created datasets comprising past APT cases. We mapped extensive historical APT attack reports to MITRE ATT&CK in various ways. Unlike the raw log datasets that were generated in a particular network, our datasets are well-suited to any kind of network setting. Therefore, BAN trained on our datasets is also independent of the network environment, maximizing its usefulness.
- We implemented a fully automated data-driven learning process. Both the structure and parameters of the Bayesian network can be learned automatically. Through experiments, we find a suitable learning algorithm and parameters for our model and propose a new learning method based on the impact-based weighted score function.
- As a result of evaluating the model with the prepared dataset, it showed an f1-score of 0.628 at the next attack prediction and 0.606 at the next defense prediction. Also, in terms of predicting the ultimate goal attack and defense, it showed an f1-score of 0.565 at the goal attack prediction and 0.635 at the goal defense prediction.

II. RELATED WORK

A. ATTACK PREDICTION IN CYBER SECURITY

Over the past few decades, many methods have been explored to defend against cyberattacks. However, as cyberattacks become more sophisticated and rapidly evolve, it is difficult to detect attacks and preemptively defend them. Thus, the defenders have to predict the next attacks and prepare proactive defense strategies. For this reason, various cyberattack prediction studies have been actively studied.

Among these attack prediction approaches, the most representative approach is attack graph [17]. After the concept of

TABLE 1. Comparison of existing cyberattack prediction models.

Study	Approach	Attack prediction level	Training dataset	Environment independent	Data-driven learning	Defense suggestion
Hughes and Sheyner. [3]	Attack Graph	Node in Attack Graph	–	✖	✖	✖
Ramaki *et al*. [4]	Attack Graph	Node in Attack Graph	DARPA 2000, DARPA GCP 3.1	✖	●	✖
GhasemiGol *et al*. [5]	Attack Graph	Node in Attack Graph	–	✖	✖	●
Lisỳ *et al*. [14]	Game Theory	Action of Game (12 classes)	–	●	✖	●
Píbil *et al*. [15]	Game Theory	Action of Game (Select Host ID to attack)	–	●	✖	●
Farhadi *et al*. [6]	Hidden Markov Model	Class of Attack (13 classes)	DARPA 2000	✖	▲†	✖
Kholidy *et al*. [7]	Hidden Markov Model	State of System (4 classes)	DARPA 2000	✖	▲†	✖
Sendi *et al*. [8]	Hidden Markov Model	State of System (4 classes)	DARPA 2000	✖	▲†	●
Fredj *et al*. [12]	Deep Learning	Type of Attack (36 classes)	DEFCON CTF 17	✖	●	✖
Li *et al*. [13]	Deep Learning	Phase of Attack (29 classes)	HDFS, OpenStack, PageRank, BGL	✖	●	✖
Qin and Lee. [10]	Bayesian Network	Node in Attack Graph	DARPA GCP 3.1	✖	▲†	✖
Ramaki *et al*. [9]	Bayesian Network	Node in Attack Graph	DARPA 2000	✖	▲†	✖
Poolsappasit *et al*. [11]	Bayesian Network	Node in Attack Graph	–	✖	✖	●
This work	Bayesian Network	MITRE ATT&CK technique (More than 120 classes)	*Expert*, *TRAM*, *rcATT*, *MITRE* ‡	●	●	●

† The authors designed the structure of the model, while only the parameters of their model were trained using the data.
‡ Our datasets are based on past APT attack cases, while other datasets mainly consist of IDS logs. See section.V-C1

an attack graph appeared in 1998, Hughes and Sheyner [3] proposed the concept of attack prediction using an attack graph in 2003. Since then, various attack graphs have been proposed. Ramaki et al. [4] proposed a real-time episode correlation algorithm (RTECA) and used it for alert correlation and prediction of multi-step attack scenarios. They also developed a causal correlation matrix (CCM) for attack prediction. Also, GhasemiGol et al. [5] improved the prediction performance by proposing an uncertainty-aware attack graph.

The Bayesian network (BN), used extensively in various fields, was also employed for attack prediction. Most of the studies using BN added the concept of probability to the attack graph. Most representatively, [10] transformed the attack graph into a causal network and used it for low-level alert correlation and next attack prediction. In addition, Ramaki et al. [9] proposed a Bayesian attack graph. They constructed a Bayesian attack graph using low-level alerts in offline mode and applied it to predict the next attack in online mode. Poolsappasit et al. [11] proposed a risk management framework for enterprise networks using a Bayesian attack graph. This study is not only focused on attack prediction but also considers security control for risk minimization. In addition, various attack prediction methods using game theory [14], [15], HMM [6], [7], [8], deep learning [12], [13] have been proposed.

Nonetheless, there are some limitations within the current research. First, the attack prediction results are either too abstract [6], [7], [8] or newly defined by the authors [3], [4], [5]. This makes it difficult for defenders to prepare the countermeasures. Second, most studies utilize raw log datasets that are applicable only to specific network. Even more, some studies do not utilize any datasets at all. This makes their models dependent on the network environment and increases the personal involvement of the authors when constructing the models. Finally, some studies do not suggest appropriate countermeasures to their attack prediction results. This leads to the difficulty of defending against predicted cyberattacks in a real-world environment.

In contrast, in this study, we utilize self-built APT datasets to identify APT attack patterns in the real-world. In addition, we adopted MITRE ATT&CK, which is environment-independent by its design, to make our model independent of the network environment. Finally, we built a model optimized for our APT datasets by minimizing the intervention of experts and automating the learning process of the Bayesian network. We note that our model is not based on an attack graph approach. Unlike a Bayesian attack graph [11], which adds the concept of probability to the attack graph, our approach is more like utilizing a traditional Bayesian network for APT attack prediction. The comparison of current cyberattack prediction studies is described in Table 1.

B. DEFENSE FRAMEWORK IN CYBER SECURITY

Diverse studies were also proposed to suggest a defense policy according to the attack. Related methods which can suggest proper defense techniques are as follows.

First, various static defense modeling techniques based on the attack graph have been proposed. Bistarelli et al. [18] proposed the defense tree to model economically effective defense policies from the defender's point of view. The same authors further developed the CP-Defense tree and calculated

the most appropriate countermeasure by replacing it with an answer set optimization (ASO) problem [19]. Roy et al. [20], [21] proposed an attack countermeasure tree (ACT) for calculating and reducing the attack risk. The major difference between the ACT and the defense tree is that, in the defense tree, defenses can be added only to leaf nodes, but in ACT, defenses can be added to any node. Kordy et al. [22], [23] proposed a new variation of the defense tree, an attack-defense tree (ADTree). The ADTree is similar to the ACT, but while the ACT aims to calculate return on investment (ROI) and return on attack (ROA), ADTree is designed for more rigorous and general purposes.

Studies on using Bayesian networks from the defender's point of view have also been explored. Sommestad et al. [24], [25] introduced the Bayesian defense graph by adding the concept of probability to the original defense tree. The author uses the Bayesian defense graph to calculate not only the probability of an attack but also the expected damage from the attack. Therefore, the defenders can utilize the Bayesian defense graph to select an appropriate countermeasure. The authors also proposed a probabilistic relational model that enables more general and accurate modeling through follow-up studies [26].

Moreover, attack and defense modeling using Boolean logic Driven Markov Processes (BDMP) [27], [28], network defense and hardening policy building algorithm using game theory [29], [30] have been proposed.

These researches about cyberattack defense also have similar limitations as the attack graph study. These approaches are subjective because defenders must build the model manually. Also, since the suggested defenses are simple techniques and do not train a model based on data, performance can be poor in practical use.

III. MODELING APT ATTACK AND DEFENSE SCENARIO

A standard limitation of cyberattack prediction and defense studies, discussed in the previous chapter, is that they cannot precisely model the attacker's behavior. Consequently, there is a critical problem that the result of the attack prediction is too conceptual, so suitable defense techniques cannot be selected from the defender's point of view. In order to establish the APT attack prediction and defense proposal system, which is the ultimate goal of this study, it is necessary to solve this issue. Hence, this chapter will review the methods of modeling attackers' behaviors and defenders' defense techniques.

A. MODELING APT ATTACK

Many methods for modeling existing cyberattacks have been proposed [31]. APT attack modeling does not differ much from cyberattack modeling techniques. However, because APT attacks use more advanced strategies over a long period, more advanced modeling techniques are being studied [32], [33].

The cyberattack modeling technique widely used in both academia and industry is Cyber Kill Chain [34], [35] developed by Lockheed Martin. However, although it can effectively model the stages of an APT attack, there is still a limitation that there is no explanation about which attack techniques can be utilized in each stage.

As mentioned in the introduction, it is essential to predict an attacker's behavior precisely. Therefore, a more detailed modeling framework than the cyber kill chain is needed. For this reason, among various techniques for modeling APT attacks, the framework selected in this study is MITRE ATT&CK® [16], [36], [37].

1) MITRE ATT&CK®

MITRE ATT&CK® is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's attack lifecycle and the platforms they are known to target [16]. MITRE ATT&CK® is based on the concept of tactics, techniques, and Procedures (TTP) [38] defined by NIST. Therefore, the items of MITRE ATT&CK® consist of tactics and techniques, excluding procedures that are too specific in TTP.

Tactics represent the ''why'' of an ATT&CK technique or sub-technique. It is the adversary's tactical objective: the reason for performing an action [16]. The MITRE ATT&CK framework consists of 14 tactics, from the *TA0043 (Reconnaissance)* tactic to the *TA0040 (Impact)* tactic.

Techniques represent ''how'' an adversary achieves a tactical objective by performing an action. Techniques may also represent ''what'' an adversary gains by performing an action [16]. Each technique is associated with one or more tactics. Sub-techniques further break down behaviors described by techniques into more specific descriptions of how behavior is used to achieve an objective [16]. The latest version at the time of this research, MITRE ATT&CK® version 10, consists of a total of 188 techniques and 379 sub-techniques.

Since MITRE ATT&CK® can model even detailed attack techniques, it is being used as a standard worldwide. Moreover, many government agencies and cyber security companies are using this ATT&CK framework to analyze the APT attacks and apply it to their security software [39]. Also, MITER ATT&CK is regularly verified and updated by numerous cyber security experts [40], [41]. Through this updating process, the latest attack TTPs are added, and existing TTPs are also re-validated to increase reliability. Because of the popularity, peer review, and up-to-dateness of MITRE ATT&CK, we chose MITRE ATT&CK as the attack modeling framework for this study. Also, it is impractical to provide more detailed attack modeling and prediction due to a lack of available data and a standardized framework.

B. MODELING APT DEFENSE

The final goal of this study is to predict APT attacks and provide appropriate countermeasures. Therefore, it is necessary to model appropriate defenses against possible attack techniques.

Modeling countermeasures against cyberattacks have been researched and organized by many institutions. For example, NIST classified and defined countermeasures to defend against cyberattacks in the SP-800-53 document [42]. CIS developed CIS Control v8 to provide defenders with an implementable level of defense [43]. In addition, ISO/IEC defines information security standards by providing ISO/IEC 27000 series [44]. Microsoft has also documented the security controls available in its cloud product; Azure [45].

1) MITRE FRAMEWORKS

MITRE, which developed the MITRE ATT&CK®, researched cyber defense techniques and created a framework. The first framework developed is ATT&CK Mitigation [46] included in ATT&CK Framework. ATT&CK Mitigation roughly described the countermeasures against the ATT&CK technique. As the MITRE ATT&CK® is updated, it is also periodically updated.

The following defense framework that emerged is MITRE Shield™ [47]. MITRE Shield™ is a publicly accessible knowledge base of active defense tactics and techniques based on real-world observations [47]. MITRE provides a correspondence relationship between MITRE ATT&CK®'s technique and MITRE Shield™'s technique to induce the defender to respond appropriately to a specific attack. However, the techniques of MITRE Shield™ have a disadvantage in that it is difficult to be used because there are too many abstract descriptions. To resolve these drawbacks, the newly developed framework after the abolition of Shield is MITRE Engage™ [48], [49].

MITRE Engage™ [48], [49] was developed to solve existing problems based on the MITRE Shield™. However, unlike the Shield, Engage focuses on denial, deception, and adversary engagement. Furthermore, unlike Shield, which is composed of tactics and techniques in the same way as MITRE ATT&CK®, Engage is composed of Goal, Approach, and Activity. In addition, each item is divided into strategic actions and engagement actions. MITRE Engage™ also provides a correspondence relationship with MITRE ATT&CK® techniques, enabling effective attack response.

Apart from Shield and Engage, MITRE D3FEND™ [50], [51] was also developed. The most significant difference between the previous three defense frameworks and DEFEND is that ATT&CK and DEFEND are mapped through ontology-based artifact analysis [51]. In addition, the reliability of the framework was increased by providing a detailed reference for the defense technology.

Since all of the frameworks introduced above provide mapping with ATT&CK, all frameworks can respond to the attack prediction result. However, the specific implementation plan of the proposed defense technology is beyond the scope of this study. Therefore, in this study, the Mitigation framework, which has been verified for the longest time, is used for convenience.

IV. BAYESIAN NETWORK

This section briefly reviews a Bayesian network, which is the main machine learning model in this study. Bayesian network is a probabilistic graphical model that has been widely used and verified in many fields for a long time. The main idea of the Bayesian network is to represent the conditional probability relationship more intuitively by using the directed acyclic graph (DAG). By definition, random variables are represented as nodes, and the relationships between the variables are represented as the edges. By composing such a model, it is possible to identify the overall relationships between variables and infer other variables when some variables are observed.

A. DEFINITION

A Bayesian network is mathematically defined as follows.

Definition 1 (Bayesian Network): Bayesian network (BN) is a triple $B = (X, G, \Theta)$ where

- Random Variables: X is a set of random variables in the domain.
- Graph: $G = (V, E)$ is a directed acyclic graph with nodes $V = \{X_1, \ldots, X_n\}$ and edge E representing direct dependencies between the variables.
- Parameters: Θ is a parameter of Bayesian network in the form of conditional probability table or conditional prabability distribution. More formally, Θ encodes the parameters $\{\theta_{ijk}\}_{i\in 1\ldots n, j\in T_{\Pi_{X_i}}, k\in T_i}$ of the network, where

$$\theta_{ijk} = P(X_i = x_{ik} \mid \Pi_{X_i} = w_{ij}). \tag{1}$$

  Π_{X_i} denotes the set of parents of X_i in G, $J_{\Pi_{X_i}}$ denotes the joint domain of the variables in Π_{X_i}, x_{ik} is the k-th value of X_i and w_{ij} is the j-th configuration of Π_{X_i}

Based on this definition, the joint probability distribution can be expressed in a straightforward form of factorizing conditional distributions.

Theorem 1 (Joint Probability Distribution in Bayesian Network): In the Bayesian network, a unique joint probability distribution over X given by

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \Pi_{X_i}). \tag{2}$$

B. LEARNING

As we can see from the definition of the Bayesian network, we need to find out X, G, and Θ to create BN. Among them, the set of random variables X is easily determined according to the problem definition. Also, the most intuitive way to create a graph G is to find the edges E by identifying associations between random variables X by a domain expert. In addition, parameters Θ can also be calculated or assigned using the knowledge of an expert.

However, since this method requires significantly skilled experts and is based on subjective expert opinions, the performance can be poor in actual data. To solve these problems, most practical studies train the Bayesian network based on the collected datasets. The collected dataset is defined as follows.

Definition 2 (Dataset): Dataset $D = \{D_1, \ldots, D_m\}$ is a set of data $D_j = \{D_j^1, \ldots, D_j^n\}$ where D_j^i is a value of X_i in data D_j

When a dataset is given, a Bayesian network's learning process is expressed as follows.

$$\underbrace{P(G, \Theta \mid D, X)}_{\text{learning}} = \underbrace{P(\Theta \mid G, D, X)}_{\text{parameter learning}} \cdot \underbrace{P(G \mid D, X)}_{\text{structure learning}} . \tag{3}$$

As seen in Eq. (3), the learning process primarily consists of structure learning and parameter learning.

1) STRUCTURE LEARNING

Definition 3 (Structure Learning of Bayesian Network): Given a dataset D and random variables X, *Structure learning* of Bayesian network consists in finding a most plausible network structure G^*.

$$G^* = \underset{G}{\operatorname{argmax}}\, P(G \mid D, X) . \tag{4}$$

The structure learning techniques of the Bayesian network, which have been actively studied and applied to many problems, are divided into three main methods: 1) score-based learning, which defines evaluation criteria for graphs and explores various graphs. 2) constraint-based learning, which directly identifies the relationship between nodes by performing the independence test. 3) hybrid learning, which combines score-based learning and constraint-based learning.

Firstly, in score-based learning, a score function $\mathrm{S}(G \mid D)$ is defined to evaluate the graph structure [52]. As a score function, likelihood is generally used as a basic metric based on information theory, and the score is penalized according to the model complexity. There are some representative examples of such a score function, include Akaike Information Criterion (AIC) [53], Bayesian Information Criterion (BIC) [53], factorized normalized maximum likelihood score (fNML) [54], etc. These score functions utilize the log-likelihood, which is shown in Eq. (5)

$$\log(P(D|G)) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) , \tag{5}$$

where:

n = Number of random variables X ($|X|$)
r_i = Number of states of X_i ($|X_i|$)
q_i = Number of possible configurations of the parent set Π_{X_i} of X_i ($|\Pi_{X_i}|$)
N_{ijk} = Number of instances in the data D where the variable X_i takes its k-th value and the variables in Π_{X_i} take their j-th configuration
$N_{ij} = \sum_{k=1}^{|X_i|} N_{ijk}$.

Also, the score function can be defined as Eq. (6).

$$\mathrm{S}(G \mid D) = \log(P(D|G)) - \phi(N)||G|| , \tag{6}$$

where:

$\phi(N)$ = Regularization function
$||G||$ = The number of parameters in the graph G.

In the likelihood-based score function, the parameters of the graph are calculated through the MLE method. As can be seen from Eq. (6), the likelihood-based scores have the term which maximizes the likelihood and the regularization term ($\phi(N)||G||$) for decreasing the model's complexity. Depending on the regularization term, it is AIC when $\phi(t) = 1$, and BIC when $\phi(t) = \log(t)/2$.

Also, there is a BD (Bayesian Dirichlet) score family to which the Bayesian learning concept is applied. These score functions include Bayesian Dirichlet equivalent score (BDe) [55], Bayesian Dirichlet equivalent uniform score (BDeu) [56], Bayesian Dirichlet sparse score (BDs) [57], K2 score [58]. The equation of BDe, the most representative BD score, is as follows.

$$\mathrm{S}_{\mathrm{bde}}(G|D) = \log(P(G)) + \sum_{i=1}^{n} \left[\sum_{j=1}^{q_i} \left[A + \sum_{k=1}^{r_i} B \right] \right] , \tag{7}$$

where:

$A = \log\left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})}\right)$
$B = \log\left(\frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})}\right)$
η_{ijk} = The hyperparameters for the Dirichlet prior distribution
$\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk}$
$\Gamma(.)$ = Gamma function .

In Eq. (7), there is no problem even if $P(G)$ is assumed as a uniform distribution, so the term $log(P(G))$ is not considered when calculating the actual score. In Eq. (7), if $\eta_{ijk} = 1$, it becomes K2 score [58], and if the prior distribution of the graph is assumed as uniform distribution, it becomes BDeu score [56].

After defining the score function, score-based learning methods search for the graph structure space and select the graph structure with the highest score. Algorithms for searching for the graph have been proposed: from greedy techniques, such as simple hill climbing [59] and Tabu search [60], to heuristic algorithms, such as genetic algorithm [61] and simulated annealing [62]. Also, innovative method, such as orderMCMC [63], searches for the order space of random variables without directly exploring the graph structure.

Secondly, the constraint-based learning methods conduct the independence tests, such as the Chi-squared test, to identify the relationship between nodes. Constraint-based learning algorithms have been developed to minimize the number of required independence tests. Examples of the constraint-based learning algorithm include PC [64], Grow-Shrink [65], and Incremental Association [66].

Finally, the hybrid learning algorithm is a structure learning algorithm that properly harmonizes constraint-based learning and score-based learning. For instance, Max-Min Hill-Climbing [66] and Restricted Maximization [67] have been proposed.

2) PARAMETER LEARNING

Definition 4 (Parameter Learning in Bayesian Network): Given a dataset D, random variables X and the network structure G, *Parameter learning* of the Bayesian network consists in estimating the optimal parameters Θ^*.

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} P(\Theta \mid G, D, X). \tag{8}$$

The most straightforward and intuitive parameter learning technique is maximum likelihood estimation (MLE) [68]. In the case of MLE, the parameters that maximize the likelihood are selected. In other words, it is a method of choosing the parameters that describe the data most accurately. Expressing MLE as a simple equation is equivalent to Eq. (9).

$$\Theta^*_{MLE} = \underset{\Theta}{\operatorname{argmax}} P(D \mid \Theta, G, X). \tag{9}$$

Unlike MLE, which learns parameters based on the likelihood, there is another parameter learning technique based on Bayes theorem [69]. These techniques are often called Bayesian learning or Bayesian parameter estimation. In MLE, parameters are treated as one fixed value, whereas in the Bayesian approach, parameters are treated as random variables with a prior distribution. Therefore, Bayesian approaches calculate the posterior distributions of parameters based on the prior distributions and given data. However, if the parameters are considered as distributions, the computation complexity for inference increases exponentially. Thus, the point estimation method to solve this problem has been proposed, called maximum A posteriori (MAP) [70]. MAP selects the fixed parameters that maximize the posterior distribution. MAP can be expressed in a simple equation as follows.

$$\Theta^*_{MAP} = \underset{\Theta}{\operatorname{argmax}} P(\Theta \mid G, D, X). \tag{10}$$

C. INFERENCE

If the Bayesian network is trained based on data or expert knowledge, it can be used in real problems. Generally, we can infer unobserved variables based on observed variables (evidence). There are two main inference methods in a Bayesian network: exact inference and approximate inference.

Exact inference is a method that infers the probabilities of the remaining nodes by conditioning the evidence. The intuitive exact inference technique is belief propagation [71]. Belief propagation efficiently calculates marginal distribution by introducing the concept of message. Also, there is variable elimination [72], which is widely known as a straightforward algorithm. Variable elimination is a technique that removes uninterested variables through summation. Finally, the clique tree propagation [73] transforms the original Bayesian network into a tree form that is easy to apply the belief propagation.

Although these exact inference techniques are designed to calculate the joint distribution efficiently, their time complexities are relatively high. Therefore, approximate inference techniques that take advantage of the time complexity while giving up inference accuracy have been actively developed.

The approximate inference is a technique that sacrifices the accuracy of inference for fast computation. A typical approximate inference is a technique of sampling unobserved events from the constructed BN. As the sampling method, simple importance sampling or an advanced sampling technique, such as Monte Carlo Markov Chain (MCMC), can be employed [74]. Another popular approximate inference technique is variational inference [75]. The core idea of the variational inference technique is transforming the inference problem into an optimization problem. That is, we can create a distribution that approximates the distribution we want to infer and then solve the optimization problem. This process allows us to make the approximate distribution as similar to the original distribution as possible. Compared to the sampling-based methods, these methods do not guarantee the global optimum but have the advantage of being more scalable.

V. BAYESIAN ATT&CK NETWORK

In this paper, we propose a Bayesian ATT&CK Network (BAN) that predicts the next attack based on the Discrete Bayesian network. Nodes of BAN consist of TTPs of MITRE ATT&CK Framework. In this study, MITRE ATT&CK version 10.0, the latest version at the time of the study, was used. To construct the BAN, we properly use the knowledge of cybersecurity experts and historical attack data. First, we analyzed past attack cases and MITRE ATT&CK Framework to identify the fundamental relationship between ATT&CK techniques (Section V-B). For dataset collection, publicly available APT reports are collected, and the ATT&CK techniques noted in each report are extracted and labeled in various ways (Section V-C1). After preparing the dataset, collected datasets are preprocessed (Section V-C2), and the structure of the BAN is trained by structure learning using dataset and expert knowledge (Section V-C3). When graph G is determined through structure learning, parameter learning is also executed using the collected dataset to calculate the conditional probability table (CPT) of BAN (Section V-C3). If the entire learning is finished, it can be used to predict the actual attack. When the APT attacks are detected through the detection system, this attack information is passed into the trained model. And then, the model can predict the next attacks through an inference algorithm of BN (Section V-C4). Finally, by suggesting countermeasures to defend against predicted attacks, BAN helps the defender respond proactively

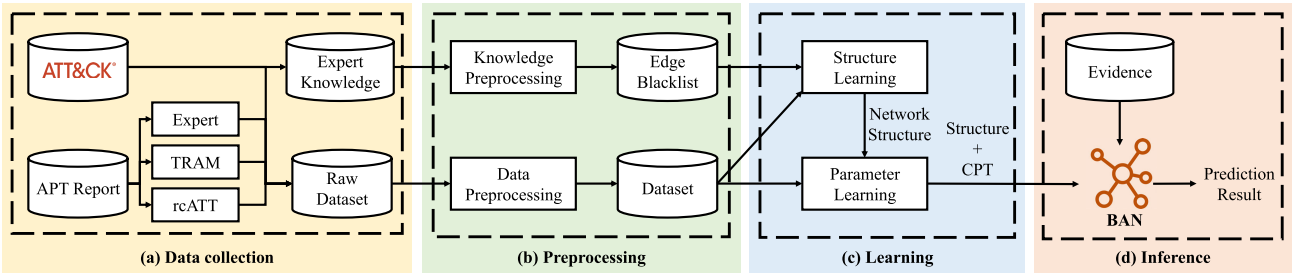

FIGURE 1. Attack prediction using BAN. The overall process consists of four steps: (a) collecting data from multiple sources and leveraging domain knowledge for model improvement, (b) preprocessing for proper input format, (c) training BAN, and (d) inference with the trained network.

to the future attacks. This way, BAN can be utilized as an effective APT defense framework in the enterprise network. (Section V-D) The figure summarizing the overall process of BAN is shown in Fig. 1.

A. WHY BAYESIAN NETWORK

The most recent SOTA machine learning models are definitely deep learning models. So naturally, deep learning models are expected to outperform the other machine learning models in APT attack prediction. However, in this paper, the Bayesian network is used because of several reasons below.

First, collecting sufficient data to train deep learning models is nearly impossible. As described above, to the best of our knowledge, we could not find a public dataset that transforms past APT attack cases into ATT&CK technique to date. Therefore, data labeling was conducted based on the attack analysis report, but only 1431 data samples were finally obtained. In addition, the usable dataset becomes even smaller when they are preprocessed. The main reason why there are few available datasets is that APT attacks occur rarely, and attacked organizations usually do not want to disclose their detailed incident report to the public. Therefore, the amount of reports published to the public is tiny. Moreover, many reports analyze the same incident, so duplicates exist. Although this problem can be solved after collecting more APT reports later, it was judged that it is challenging to train the deep learning models with only 1431 data at this time. When using deep learning, we probably cannot get the desired performance if the dataset is small [76]. On the other hand, the Bayesian networks have been building models with relatively few datasets for a long time [77], [78], [79].

Second, The dataset contains samples of missing data. Indeed, the methodology of data labeling for the APT reports was sufficiently verified by experts to increase the trustworthiness. However, there is no way to improve the completeness of the collected report itself. In fact, many reports omit the intermediate stage of the attack, so labeled data often omits the intermediate process. Fortunately, methods for learning BN from these missing data have been studied and verified for a long time [67]. Therefore, these techniques can overcome the limitations of data to some degree.

Third, in the case of BN, the knowledge of experts can be easily reflected in the model. As described earlier, security experts can identify the relationships between TTP using detailed descriptions and references. In situations where data is insufficient, this knowledge can help to improve model performance. In the case of BN, it is easy to incorporate expert knowledge because the model consists of the factors that have to be analyzed as nodes and the relationships between the factors as edges.

Lastly, in the case of BN, the interpretability is exceptionally high compared to deep learning. Since the model structure is not complicated and each node has its meaning, humans can understand the structure of the BN. This advantage can be utilized in the model validation process, and associations or causal relations between random variables can be identified.

B. EXPERT KNOWLEDGE

As mentioned earlier, one of the advantages of BN is that prior knowledge can be applied to the model. If the dataset is insufficient and the patterns are unclear, this prior knowledge can have a significant effect [80]. Therefore, to build the BAN, we identified the relationship between the nodes constituting the BAN, that is, MITRE ATT&CK techniques.

The relationship between MITRE ATT&CK TTP can be inferred in various ways. Several previous studies tried to identify the relationship between TTP based on the data, but the identifiable relationship was very limited [81]. Thus, in this study, we identified the relationships between ATT&CK TTP based on the order of the APT attack and the required privilege for the attack. We identified direct relationships like TTP A affect another TTP B. These relationships can be utilized later in structure learning.

1) RELATIONSHIPS BASED ON THE ORDER OF ATTACK

The first and most intuitive relationships are established using the order of attack. The APT attack process has a partly explicit order.

For example, in most APT attacks, *TA0001 (Initial Access)* techniques come first. Afterward, the attacker completely penetrates the target system through techniques such as *TA0002 (Execution)*, *TA0003 (Persistence)*, and *TA0004 (Privilege Escalation)*. From then on, the attackers typically collect the information they need through *TA0009*

(Collection), *TA0011 (Command and Control)* techniques. Finally, the ultimate goals of the attack are *TA0010 (Exfiltration)* and *TA0040 (Impact)*, which are attack actions such as information stealing or denial of service. This general order is commonly shown in numerous APT modeling frameworks [33]. Based on this order, we defined five steps of the APT attack: Initial Access, Code Execution, System Penetration, Data Collection, and Goal Achievement. Table 2 summarizes the order of attacks based on ATT&CK tactic.

TABLE 2. Attack step of tactics defined in ATT&CK.

Step ID	Step Name	Tactic ID
1	Initial Access	*TA0001*
2	Code Execution	*TA0002*
3	System Penetration	*TA0003*, *TA0004*, *TA0005*, *TA0006*, *TA0007*, *TA0008*, *TA0011*
4	Data Collection	*TA0009*
5	Goal Achievement	*TA0010*, *TA0040*

As seen in Table 2, if the order between ATT&CK tactics is unclear, it is regarded as the same attack stage. Note that we consider all MITRE ATT&CK Tactics, as there are only five steps, but each step can include multiple tactics. There may be some exceptional attack cases, but most attacks follow the order of Table 2. Therefore, we assumed that relationships that reverse the order of attacks are unlikely.

2) RELATIONSHIPS BASED ON THE REQUIRED PRIVILEGE

One of the key prerequisites that influence the adversary's decision on the next attack is the privilege acquired on the target system. MITRE ATT&CK techniques may require a specific or higher level of privilege than the adversary already gained. In the latter case, the adversary should precede some of the techniques under *TA0004 (Privilege Escalation)* tactic to acquire the higher privilege. For instance, *T1136 (Create Account)*, a technique that creates accounts to maintain the persistence of attack, requires Administrator privilege to create users. To implement that technique, the adversary must precede a technique under *TA0004 (Privilege Escalation)* tactic that can gain Administrator privilege.

MITRE ATT&CK's official description [36] contains a list of minimum privileges each attack technique requires. In MITRE ATT&CK, the required privileges are divided into five types: User, SYSTEM, Administrator, Remote Desktop Users, and root. Among these five types of privileges, User privilege can be acquired through the techniques under *TA0002 (Execution)* tactic. Also, the attackers can gain Remote Desktop Users by typical techniques but not requires escalation of privilege. Therefore, only the techniques that require administrator-level privileges, such as SYSTEM, Administrator, and root, should be acquired by *TA0004 (Privilege Escalation)* techniques before the former technique is executed. Consequently, In this paper, we consider that the techniques under *TA0004 (Privilege Escalation)* tactic should precede the techniques which require more than the user-level privilege (i.e., SYSTEM, Administrator, root). On the other hand, we assumed that relationships from the techniques that require admin-level privilege to *TA0004 (Privilege Escalation)* techniques are unlikely.

C. OVERALL PROCESS OF BAN

1) DATA COLLECTION

A prerequisite for constructing a high-performance model is preparing a high-quality dataset for training. However, unfortunately, to the best of our knowledge, there is still no standard dataset about past APT attack cases. While there are existing datasets [82], [83], [84] that simulate APT attacks, they have several limitations. First, they are based on host and network logs, so they are dependent on a certain network environment. Also, these datasets are generated by conducting simulated attacks on a virtual network, so there are unavoidable differences between real-world network environments and actual APT attacks. In addition, they are generated with only two or three attack scenarios at most, so they cannot represent a diverse set of APT attack patterns. Therefore, in this study, we built the best datasets to represent diverse real-world APT attack cases that have really occurred in the past.

For training BAN, we need the data that summarizes which ATT&CK techniques were used in a specific APT attack case. Unfortunately, we cannot collect and analyze the raw data about previous APT attacks. Instead, we collected the data already managed by MITRE and generated the data by analyzing the investigation report of past APT cases.

First, The most straightforward data is reference contents posted on the MITRE ATT&CK website [36]. The technique description page of the MITRE ATT&CK website describes the technique in detail. It also includes examples of actual APT cases that exploited that technique in the reference section. There are diverse types of reference documents, but in most cases, they are incident reports analyzing the APT attacks. Therefore, the technique ID mapping corresponding to the specific APT case can be acquired by extracting the technique list mapped for each reference. Hence, we collected all the references and the technique ID mapping from the website to construct the *MITRE* dataset.

Second, the technique ID was manually labeled by directly analyzing the APT incident reports. In this paper, we constructed the data based on reports managed by CCC (APT & Cybercriminals Campaign Collection) [85]. The CCC contents consist of blog posts, reports, and presentations of APT campaigns. The collection duration of reports is from 2006 to 2022. For constructing the dataset, a total of 1,143 reports were manually analyzed by cyberattack experts, and ATT&CK technique ID was labeled to the collected reports. In the case of manual labeling, both tactic information and technique information were extracted because experts can determine for what purpose the attack was executed. As criteria for manual labeling, we selected two guidelines [86], [87] published by Cybersecurity & Infrastructure Security Agency (CISA) and MITRE, respectively. Labeling work was performed by security experts familiar with CTI,

and these experts were thoroughly educated on our guidelines. In addition, we periodically verified that the manual labeling criteria were being fulfilled. In the rest of this paper, we named this dataset as *Expert* dataset.

Finally, two more datasets were generated using the natural language processing model. Reports collected from the CCC were automatically labeled using rcATT [88], [89] and TRAM [90]. rcATT and TRAM are tools that automatically tag ATT&CK TTP using a machine learning model by receiving report contents as input. rcATT tags both tactic and technique, whereas TRAM only tags technique and sub-technique. Unfortunately, both rcATT and TRAM were originally implemented with the previous version of MITRE ATT&CK. Therefore, we partially modified the source code, and the model was re-trained based on MITRE v10.0.

TABLE 3. Data format from different labeling methods.

Labeling	Format	Example
MITRE	*(Sub)techniqueID*	[*T1590, T1590.005*]
Expert	*tacticID.(Sub)techniqueID*	[*TA0002.T1053.005*]
rcATT	*tacticID* or *techniqueID*	[*TA0003, T1547*]
TRAM	*(Sub)techniqueID*	[*T1036, T1021.003*]

A total of 4 datasets prepared in this paper have slightly different labels depending on the type. Each dataset's detailed format and examples are expressed in Table 3. First, in the case of the *MITRE* data set, only the technique information was labeled because the data was collected from the technique and sub-technique page of the MITRE ATT&CK official website. Tactic information is omitted because there is no reference on the tactic description page of the official website. In contrast, in the case of the *Expert* dataset, which was labeled manually, it was possible to determine the intention of the specific attacks by analyzing the detailed descriptions of the report. Therefore, the label format is composed of *(tactic_ID).(technique_ID)* format. Finally, in the case of *rcATT* and *TRAM* datasets, the labeling format is determined by the original labeling model. In the case of the *rcATT* dataset, labels exist in the form of *(tactic_ID)* and *(technique_ID)* separately. Since rcATT cannot extract sub-technique information, the *rcATT* dataset has no sub-technique label. In the case of the *TRAM* dataset, labeling is performed in the format of *(Techinque_ID)* except for tactic information. Also, unlike rcATT, TRAM can label the sub-technique accurately.

The data sources of our datasets all contain the latest APT attack cases. In the case of the *MITRE* dataset, its data source, all references in MITRE ATT&CK website [36], consists of data through 2022. In addition, the CCC repository [85] contains APT attack reports from 2006 to 2022. Therefore, the *Expert*, *TRAM*, and *rcATT* datasets generated from these reports contain APT attack data from 2006 to 2022.

Note that all of the four data were derived from APT attack reports that have occurred in the past. Hence, they are highly indicative of real-world scenarios. In addition, due to the frequency of real-world APT attacks, there are limitations to collecting a lot of data. However, we have collected all the APT reports we could find through extensive research, so we believe our dataset is sufficiently generalizable to real-world APT attacks. This is supported by the fact that the *Expert* dataset covers all MITRE ATT&CK tactics and achieves a technique coverage of 90.96%.

2) DATA PREPROCESSING

As mentioned earlier, since the dataset was created by referring to the website and the analysis reports, there were cases where the data sample was abnormal if the original report was incomplete. Therefore, in order to improve model performance, useless information was preprocessed from the dataset.

First, data with less than five labeled TTP were excluded from the dataset. Because the collected reports were prepared by different organizations, some organizations described the attack process too simplistically. The data from these reports were excluded from the learning process because it was judged that only a partial process of the APT attack was described.

Next, we deleted all TTP that cannot be detected or predicted in real-world usage from the dataset. Some TTP during the attack procedure described in the report are practically impossible to predict or detect. For example, in the case of the *TA0042 (Resource Development)* tactic and its techniques, it is genuinely impossible to detect and predict an attack because attackers develop the attack resources before the attack in their own environment. In addition, in the case of *TA0043 (Reconnaissance)* and its techniques, it is virtually impossible to predict because it proceeds in the significantly early stage of the attack. These TTP were also classified as PRE attack category in MITRE and specified as the pre-stage of the attack. Therefore, all of these PRE attack TTP were deleted from the dataset.

Finally, the dataset can be converted into various versions through preprocessing. BAN nodes can be configured in diverse ways. For example, most simply, a node can be composed of only techniques information while ignoring information on tactics and sub-techniques. Another method is that tactic and technique information is composed of independent nodes. Furthermore, two pieces of information can be linked to form a single node like *tactic_ID.technique_ID*. Therefore, since there are different versions of node configuration, format conversion between data labels is necessary. Converting the format of labels is simple. Naturally, it is impossible to transform in the direction of increasing the amount of information. For example, it is impossible to infer tactic information with only a technique label. However, vice versa, converting the data of format *tactic_ID.technique_ID* into format *tactic_ID* and *technique_ID* can be performed by simply splitting the label into two. Also, if we want to omit specific information like a sub-technique, the label is replaced with the technique ID to which the sub-technique belongs.

3) LEARNING

In order to learn BAN, the first step is to select the graph's structure. In this paper, the score-based learning method is used among the structure learning methods mentioned in Sec IV-B1. We have also tried other learning methods. However, since the number of datasets is not enough compared to the number of random variables constituting the nodes of BAN, constraint-based learning and hybrid learning cannot infer the connections between nodes sufficiently.

As search algorithms used in score-based learning, hill climbing [59] and Tabu search [60] were employed. In addition, nonparametric bootstrapping [91] technique was used to increase the model's reliability during the training process. Multiple models are trained in parallel with the resampled data through the bootstrapping technique, and the results are combined into one model using the model averaging [92] technique. Also, to escape from the local optimum, structure learning can be started with not only an empty graph but also a random graph, which is generated by Ide's and Cozman's DAG generation algorithm [93]. Moreover, to restrict the model's complexity, the maximum number of parent nodes was fixed to 10.

To build an efficient score-based learning algorithm, we tested various existing score functions. Especially, the most widely used score function, BIC [53] and BDe [55] were mainly tested. In addition, we designed the new score functions to focus on the attack prediction problem effectively. The new score functions were proposed based on the intuition that it is adequate to defend the attacks with a high impact from the defender's point of view. For example, in the initial stage of the attack, *TA0001 (Initial Access)*, the impact of the attack is low because damaging actions such as data collection or data leakage have not been completed yet. On the other hand, *TA0040 (Impact)* techniques, usually the last stage of the APT, are the attack actions to obtain the attacker's final target, so the attack's impact is considerable. Therefore, it is more effective for the defender to predict and defend the attack techniques belonging to *TA0040 (Impact)* rather than the attack techniques of *TA0001 (Initial Access)*. We then designed a score function that gives a high score to the model that predicts high-impact attacks after qualitatively assigning the impact of each tactic.

TABLE 4. Impact score of tactics.

ID	Name	Impact
TA0001	Initial Access	0
TA0002	Execution	1
TA0003	Persistence	2
TA0004	Privilege Escalation	2
TA0005	Defense Evasion	2
TA0006	Credential Access	2
TA0007	Discovery	2
TA0008	Lateral Movement	2
TA0009	Collection	3
TA0011	Command and Control	4
TA0010	Exfiltration	5
TA0040	Impact	6

The impact score of each tactic was defined based on the order and importance of the attack. As we defined the order of attack in Table 2, we assigned a high impact score to the attack tactics which occur later. Even though both the initial and latter steps of APT attacks are critical, the attacker's ultimate goal generally tends to be outlined in the latter steps, e.g., *TA0010 (Exfiltration)* and *TA0040 (Impact)*. Therefore, it is important to prioritize and predict these phases to successfully prevent the attacker from achieving their objective. Furthermore, we subdivided the impact score based on the importance of the attack. For example, In the third step (System Penetration), we gave a higher score to *TA0011 (Command and Control)* because it has been observed to be more prevalent and destructive in real-world cases compared to other tactics in the third step [1]. Moreover, *TA0009 (Collection)* and *TA0011 (Command and Control)* were scored slightly higher than the other tactics because they are directly related to the ultimate goal of the attackers like *TA0010 (Exfiltration)* and *TA0040 (Impact)*. For example, *TA0009 (Collection)* is a process that must be performed immediately prior to *TA0010 (Exfiltration)* and *TA0011 (Command and Control)* is also essential for many attacks beyond *TA0010 (Exfiltration)* and *TA0040 (Impact)*. At last, *TA0010 (Exfiltration)* and *TA0040 (Impact)* belong to the same step (Goal Achievement), but the importance of these tactics is different. Since *TA0040 (Impact)* includes the attack techniques that are related to the availability of the system, which is critical to the entire organization, we assigned the highest score to *TA0040 (Impact)*. The detailed impact score of each tactic is shown in Table 4.

We propose impact-based weighted score functions by modifying the existing score function. In the summation formula of the original score function, different weights are multiplied according to the impact of the node. The exact formulas for our new score function are Eq. (11) and Eq. (12).

$$\mathrm{S}_{\mathrm{wbic}}(G\,|\,D) = \sum_{i=1}^{n} \alpha^{\beta} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) - \log\left(\frac{N}{2}\right) ||G||\,, \tag{11}$$

$$\mathrm{S}_{\mathrm{wbde}}(G\,|\,D) = \log(P(G)) + \sum_{i=1}^{n} \alpha^{\beta} \left[\sum_{j=1}^{q_i} \left[A + \sum_{k=1}^{r_i} B\right]\right], \tag{12}$$

where:

α = Hyperparameter, the default value is 1.2
β = Impact score of node X_i, as shown in Table 4,

and the rest of the parameters are same as described in Eq. (6) and Eq. (7).

Eq. (11) and Eq. (12) are the results of converting the BIC score and BDE score into an impact-based weighted score function, respectively.

The expert knowledge built in Section V-B is utilized as prior knowledge in the structure learning process. The

simplest way to utilize expert knowledge in structure learning is to establish a blacklist or whitelist of edges. Edges specified as blacklists are never searched for during structure learning. On the other hand, in the case of an edge specified as a whitelist, the corresponding edge is necessarily included in the result of structure learning. In this study, The relationships identified in Section V-B are all directional relationships. Therefore, we reduced the search space of structure learning by setting all edges that reverse the extracted relationships as a blacklist. We only utilized the blacklist because the edge whitelist approach would significantly restrict the structure of the model and interfere with our data-driven learning [94].

Algorithm 1 Structure Learning of BAN

function Structure-Learning
  Input D: dataset
  Input $S(.)$: score function
  Input E_B: edge blacklist
  Input rg: using random graph or not
  Output G: structure of BAN

  if rg **then**
    $G \leftarrow$ ic-dag(X) ▷ initial random graph [93]
  else
    $G \leftarrow (X, \emptyset)$ ▷ initial empty graph
  $s_G \leftarrow S(G)$ ▷ score value of initial network
  $s_{max} \leftarrow s_G$
  while s_{max} has increased **do**
    for all possible arc addition, deletion, reversal **do**
      $G^* \leftarrow$ modified network from G
      if G^* has cycles **then**
        continue
      if G^* has any edge in E_B **then**
        continue
      $s_{G^*} \leftarrow S(G^*)$
      if $s_{G^*} > s_G$ **then** $G \leftarrow G^*$, $s_G \leftarrow s_{G^*}$
    if $s_G > s_{max}$ **then** $s_{max} \leftarrow s_G$
  return G

The algorithm explaining the entire structure learning process is the same as Algorithm 1. We selected the hill climbing algorithm as the search algorithm in Algorithm 1. However, even if Tabu search is selected as the search algorithm, there is no significant difference from the above algorithm.

After learning the structure G of the model through structure learning, the CPT of BAN has to be learned through parameter learning. As we saw in Section IV-B2, parameter learning is straightforward when G is known. We only need to decide whether to use the Likelihood-based method or the Bayesian method. In this study, parameters were learned using the MAP technique with a uniform prior. Because of the problem condition, each node of the BAN, indicating the presence or absence of an ATT&CK technique, is a binary random variable that can have a *TRUE* or *FALSE* value. Therefore, a uniform Dirichlet distribution is given as the CPT prior distribution of each node. After that, the posterior distribution is calculated based on the given data, and the parameters that maximize the posterior distribution are selected as CPT.

4) INFERENCE

BAN usually has a complex Bayesian network structure. Theoretically, the maximum number of possible nodes is the same as the number of MITRE ATT&CK techniques (excluding the PRE technique). Of course, since not all TTP appear in the dataset, the number of nodes may be less, but there were approximately 120 or more, depending on the training dataset. Therefore, exact inference, which requires much computational cost, is practically impossible. For this reason, the algorithms we can choose are limited to approximate inference methods. Thus, a simple sampling-based likelihood weighting technique was selected for the inference algorithm of BAN.

Likelihood weighting [95] uses evidence to infer the posterior distribution of the remaining nodes. In this case, evidence refers to information that we already know, that is, the attack techniques that have been detected so far. Also, the likelihood weighting method uses a sampling technique for inference. First, the remaining nodes, except the evidence nodes, are sampled according to the topological order among all BAN nodes. After sampling is completed for all nodes, we can calculate the likelihood weight using the evidence node. Finally, by repeating these sampling and calculating likelihood weights, the posterior distribution of the remaining nodes can be inferred by calculating the average of the likelihood weights. This way, we can efficiently calculate the probability of *TRUE* of all BAN nodes, and versatile predictions are possible using the results.

In order to effectively utilize the attack prediction results, it is necessary to identify defenses to defend against the attacks. Since MITRE has already compiled defense techniques that can defend against ATT&CK technique, we can determine the defense techniques against the predicted attacks by extracting the defenses mapped to the attacks.

D. OBJECTIVES OF BAYESIAN ATT&CK NETWORK

There are several ways to utilize the inference result of BAN. Typically, the next attacks can be predicted, and the attacker's ultimate goal is also predictable. In addition, based on the attack prediction result, it is possible to suggest a suitable defense method for the defender. These predictions obtained from the Bayesian ATT&CK Network should be integrated with other security frameworks, such as attack detection systems and automatic response systems. This can not only allow for a more proactive defense but also reduce the risk of incorrect predictions. A schematic illustration of each objective of BAN is shown in Fig. 2.

The upper part of Fig. 2 describes in chronological order of attack. Attacks that have already occurred are described in the upper left of the figure. This information can be used for attack prediction in the form of evidence. Our

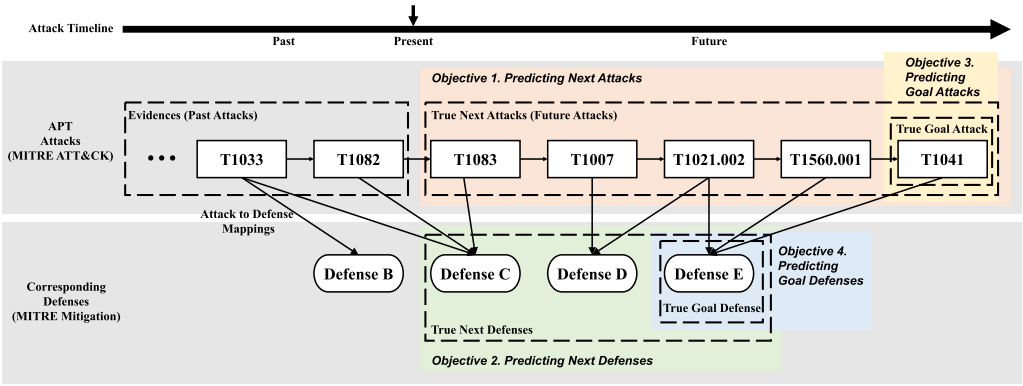

FIGURE 2. Four objectives of BAN: 1) predicting next attacks, 2) predicting next defenses, 3) predicting goal attack, and 4) predicting goal defense. Each objective aims to find countermeasures corresponding to the predicted adversarial behaviors. .

crucial objective is predicting future attacks that have not yet occurred. These future attacks are represented in the upper right of the figure. Moreover, out of these future attacks, there is an attack corresponding to the attacker's ultimate goal. The attacker's ultimate goal is defined as techniques belonging to *TA0010 (Exfiltration)* tactic and *TA0040 (Impact)* tactic because of the definition of ATT&CK framework. For example, in the figure, itT1041 (Exfiltration Over C2 Channel) corresponds to the ultimate goal technique.

As explained in Section III, each attack has defenses that can defend it. These defenses are represented at the bottom of the figure. Also, the mappings between attack and defense techniques are depicted by arrows. Furthermore, among the defenses, some defenses can defend against the ultimate attack technique, like Defense E in the figure. Thus, we named these defense techniques as the final defense goal.

1) OBJECTIVE 1: PREDICTING NEXT ATTACK TECHNIQUES

In order to effectively defend against APT attacks, a preemptive defense is essential. The most representative application objective of BAN is predicting the next attack techniques in the form of the MITRE ATT&CK technique. In the example of Fig. 2, the future attack techniques on the upper right, highlighted in orange, should be predicted using the already detected attack techniques on the upper left.

Predicting the next attacks is uncomplicated in BAN. Using the attacks that have occurred so far as the evidence, the probabilities of *TRUE* of the remaining nodes can be calculated using the inference method. After then, K nodes with the highest probability are proposed as an attack prediction result. K is a hyperparameter that is set to 5 as a default value.

2) OBJECTIVE 2: PREDICT NEXT DEFENSE TECHNIQUES

Attack prediction results alone are not helpful to the defender. In the end, the useful information for the defender is the detailed defense techniques that need to be executed instantly. In the example of the figure, information about Defense C, highlighted in green, that can defend against T1083 is required from the defender's point of view.

When an attack prediction result is derived, it is easy to find defense methods to block it. We can identify the MITRE ATT&CK Mitigation associated with each predicted attack technique and provides it to the defender. Through this, the defender can implement a concrete and realistic defense.

3) OBJECTIVE 3: PREDICTING GOAL ATTACK TECHNIQUE

Predicting the next attack is essential, but predicting the attacker's ultimate goal is even more essential. The defender can prepare more effective defenses if the attacker's ultimate goal is known. In MITRE ATT&CK, the attacker's goals are described in the *TA0010 (Exfiltration)* tactic and *TA0040 (Impact)* tactic. Therefore, we assumed the techniques belong to *TA0010 (Exfiltration)* and *TA0040 (Impact)* as goal attack techniques highlighted in yellow.

Predicting the goal attacks is similar to the next attack prediction. After receiving the evidence, BAN calculates the probabilities for the other nodes and selects only the nodes belonging to the goal technique. The K nodes with the highest probability of *TRUE* are determined as the final goal. K is a hyperparameter that can be assigned but is set to 3 by default.

4) OBJECTIVE 4: PREDICTING GOAL DEFENSE TECHNIQUE

The success of defending against an APT attack depends on blocking the attacker's ultimate goal. No matter how successful multiple attacks are in the middle of an attack, if the attacker's final goal is not achieved, it can be regarded as a success for the defender. Therefore, to effectively utilize BAN, it is necessary to propose a defense technique that can block the attacker's goal. For instance, in Fig. 2, we have to suggest Defense E, highlighted in blue, to the defender.

Because BAN can also predict the attacker's ultimate goal, it can also suggest the defense method to thwart the goal. Again, it is enough to extract the defenses corresponding to the detected goal techniques.

VI. EXPERIMENTS

The collected datasets of Section V-C1 were used to conduct the experiments. In addition, to identify the factors

involving the model's performance and find the optimal model, we specified a total of 6 research questions. We prove the effectiveness of our methods through these research questions.

Unfortunately, we have encountered obstacles in performing comparisons with previous studies due to significant differences in the level of attack prediction and dataset types used. Our model predicts up to the MITRE ATT&CK technique level, more than 120 classes, a far broader range than prior models. Further, while previous studies have utilized log datasets, we utilized APT report-based datasets, so the format of the data is significantly different. These variations prohibited direct comparisons with prior studies. Instead, we evaluated our model's performance under different scenarios and parameter settings to determine optimal configurations.

A. IMPLEMENTATION DETAILS

We implemented Bayesian ATT&CK Network in R language and Python. To implement the basic features of Bayesian network, we used R packages such as bnlearn [96] and BiDAG [97]. Other tasks, such as data collection, data preprocessing, result analysis, and model visualization, are implemented in Python 3.8. We selected appropriate Python packages like sci-kit-learn, pandas, and matplotlib.

In addition, all functions are designed to enable distributed processing to improve the speed of learning and inference. Hence, BAN can use multiple CPU cores in parallel.

B. EXPERIMENT SETTINGS

All implementations and experiments were conducted in Windows 10 64-bit environment with Intel Core i9-10980XE (3.00GHz, 36CPU) and 128GB RAM. For the stability of the experiment, we only used 30 cores.

We tested all the objectives of BAN, which are described in Section V-D, during the experiments. Since all problem settings of objectives are multi-label classification problems, the performance metrics of the model are slightly different from a single-label problem [98].

Therefore, we calculated the f1-score of each instance in the dataset and then calculated their average.

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2 * Precision * Recall}{Precision + Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{2\left|Y_i \cap \hat{Y}_i\right|}{|Y_i| + |\hat{Y}_i|},$$

where:

n = Number of samples in dataset
Y_i = True labels of the i-th sample
$\hat{Y}_i$ = Prediction labels of the i-th sample.

We used this instance-averaged (sample-based) f1-score as the default performance metric of BAN, as described above. In this study, only the f1-score was used as it provides a comprehensive measure of important metrics like precision and recall.

The assumptions for the evaluation of the four objectives are summarized as follows.

- **Predicting Next Attacks**: The last K attacks are predicted while the attack is in progress, excluding the M last attacks. At this time, K was fixed at 5, and M was different for each research question. However, if there is no mention, M was given as 5.
- **Predicting Next Defenses**: The prediction is carried out in the same scenario as the next attack prediction. However, in this case, all defenses that can defend against the predicted attacks are suggested, not just K techniques. Additionally, duplicate defense techniques have been removed.
- **Predicting Goal Attacks**: The goal attack techniques were predicted when all attacks had been finished except for the last M attacks. In this case, the number of goal attack techniques differs depending on the samples of the dataset. Therefore, there are some samples without any goal attack techniques. In practice, it makes no sense that there is no final goal of the APT. Thus, these samples were considered incomplete, so we excluded these samples from the evaluation process.
- **Predicting Goal Defenses**: The goal defense predictions were also evaluated in the same situation as the goal attack prediction. Furthermore, like Predicting Next Defense, BAN identified all defense techniques without duplicates.

In all experiments of this paper, it was assumed that the attack detection system detects all attacks. Therefore, all of the ATT&CK techniques that have been performed were detected and passed to the BAN as evidence. Performance evaluation in the environment where attack detection is imperfect was performed only in RQ6.

C. EXPERIMENTAL RESULTS

As mentioned earlier, six research questions were designed and experimented to test the performance of BAN in various ways.

All the different models trained in our experiments took 3-4 hours to train. Although it may appear to be a lengthy time, in practical scenarios of BAN, the training time is not a significant concern since the models are trained beforehand and then utilized. Furthermore, it only took approximately 20 seconds to predict a cyberattack on the trained BAN. Hence, there is no notable delay in the attack prediction when applied in real-world scenarios.

1) RQ1. HOW PERFORMANCE VARIES DEPENDING ON THE TYPE AND FORMAT OF THE DATASET?

Firstly, to determine which dataset is the best, we conducted the same experiment multiple times while changing the training and validating dataset. As for the dataset used in this experiment, 4 dataset types (*Expert*, *rcATT*, *TRAM*, *MITRE*) were prepared in 6 versions.

As shown in Fig. 3, the *MITRE* dataset showed the best performance. The *MITRE* dataset showed an f1-score of 0.614 in predicting the next attack. The other datasets, generated based

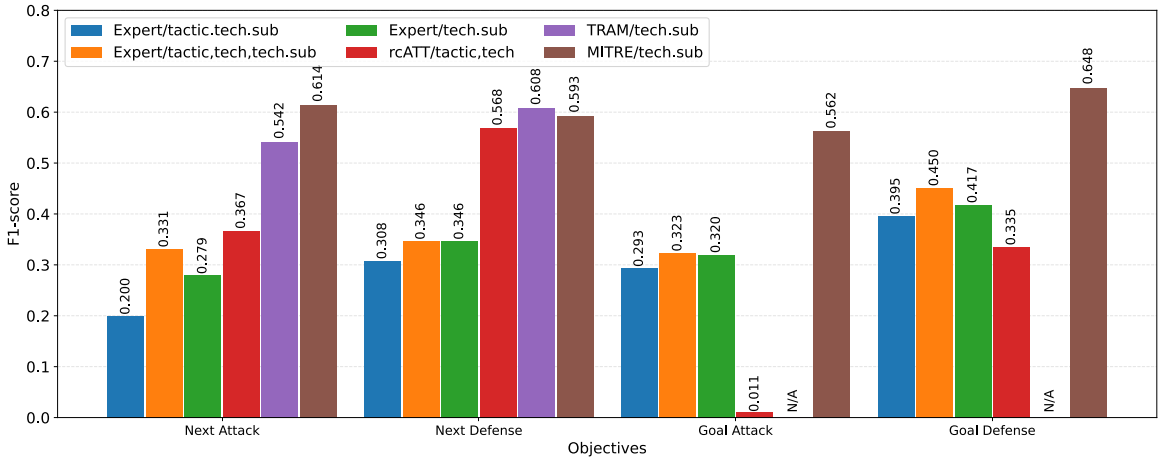

FIGURE 3. Performance comparison across various dataset types and formats.

on the APT report, performed worse than the *MITRE* dataset. The *rcATT* and *TRAM*, which are automatically labeled using the trained machine learning model, showed f1-scores of 0.367 and 0.542, respectively. In addition, the *Expert* dataset, which experts manually labeled by analyzing the reports, was transformed into various versions. First, the original data format, the "tactic.tech.sub" format, showed an f1-score of 0.200, showing disappointing performance. However, when the node structure was changed by converting the data format, it showed better performance. In the case of BAN transformed into the tactic, technique, and sub-technique node, the performance was 0.331. When tactic information was deleted, the performance was 0.279. The objective of predicting the next defense was also not significantly different from the attack prediction result. Overall, the performance of the next defense prediction was slightly improved better than the performance of the next attack prediction. However, the order of performance among the datasets did not change.

The dataset that showed the best performance in the goal attack prediction was also the *MITRE* dataset. In the goal attack prediction and goal defense prediction objectives, the *MITRE* dataset showed f1-scores of 0.562 and 0.648, respectively. The next best-performing dataset was the *Expert* dataset. In the case of the *Expert* dataset, we were able to obtain an f1-score of about 0.3 for the three converted versions. However, in the case of the goal attack prediction objective, some datasets showed poor performance. First of all, in the *TRAM* dataset, *TA0010 (Exfiltration)* and *TA0040 (Impact)* techniques are not labeled due to the fundamental design of the TRAM model. Therefore, the goal attack technique did not appear in the dataset at all, and thus the goal attack technique cannot be predicted. In the case of *rcATT*, the performance was unsatisfactory because *TA0010 (Exfiltration)* and *TA0040 (Impact)* techniques did not appear much in the corresponding dataset, due to the design of rcATT model.

In summary, the *MITRE* dataset showed the best performance in the four objectives. Note that the experiments of the rest research questions were conducted based on the *MITRE* dataset.

In 4 datasets transformed into 6 versions, the *MITRE* dataset showed the best performance in the overall objectives at most $\times 3$ more.

2) RQ2. DOES THE PERFORMANCE DEPEND ON THE STRUCTURE LEARNING ALGORITHM?

Next, we experimented to see how the performance changes according to the type of search algorithm. The testing search algorithms were hill climbing without random graphs, hill climbing with random graphs, Tabu search without random graphs, and Tabu search with random graphs. In the rest of the paper, the above four algorithms are abbreviated as *hc*, *hc_rr*, *tabu*, and *tabu_rr*, respectively. If the random graph option is enabled, graph search starts with the generated random graph, otherwise, graph search starts with the empty graph. The experimental results of RQ2 are shown in Fig. 4.

As a result of the experiment, the Tabu search algorithm generally performed the best. In the case of Tabu search, predicting the next attack, the next defense, and the goal defense showed the best performance. Also, in predicting the goal attack objective, the hill climbing algorithm showed the best performance. Unexpectedly, there was no significant performance improvement when learning from random graphs rather than empty graphs.

Although the Tabu search algorithm showed the best performance, the performance difference from the rest of the algorithms was not significant. The difference between the best and worst performing algorithms in the four objectives did not exceed 0.04. The reason seems to be that there is no significant difference between the four search algorithms since these algorithms are straightforward greedy search algorithms.

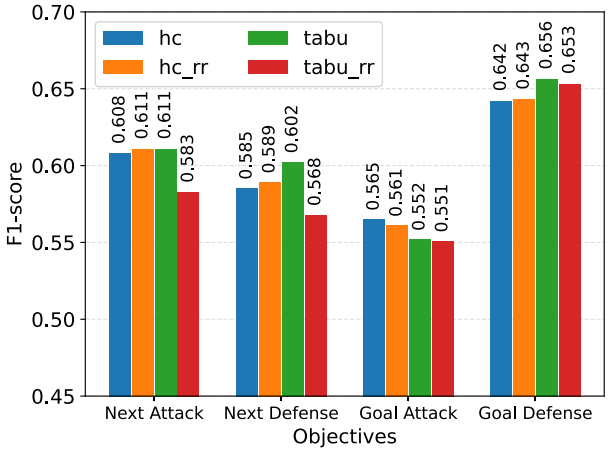

FIGURE 4. Performance comparison of different structured learning algorithms.

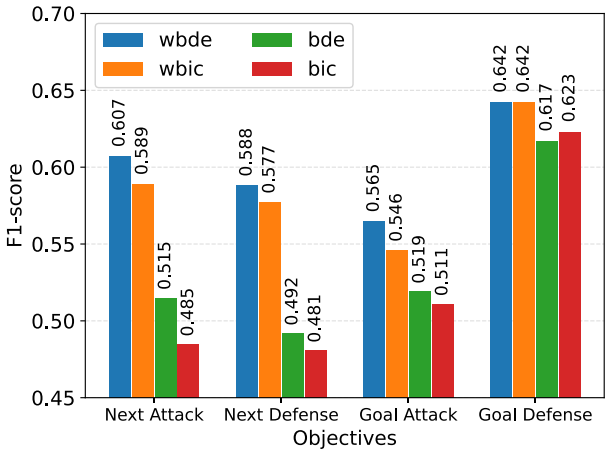

FIGURE 5. Performance comparison across different types of score functions.

The graph search algorithm did not bring any significant difference (< 0.04) in the performance.

3) RQ3. DOES THE PERFORMANCE CHANGE DEPENDING ON THE TYPE OF SCORE FUNCTION?

Next, we experimented to see how the BAN performance differs according to the score function, which is the core of score-based structure learning. We compared four score functions: Bayesian Information Criterion (*bic*), Bayesian Dirichlet equavalent score (*bde*), impact-based weighted BIC (*wbic*), and impact-based weighted BDe (*wbde*). The results are shown in Fig. 5.

Looking at the result, it was verified that the impact-based weight score functions, which are proposed in this paper, significantly improved the performance of the BAN. Both *wbde* and *wbic* showed better f1-score than the existing *bde* and *bic*. Also, in particular, *wbde* showed the best performance in 4 prediction objectives. In the next attack prediction objective, the f1-score difference between the best-performing *wbde* and the worst-performing *bic* was about 0.122. This difference was slightly reduced in other objectives, but the score functions proposed in this paper generally showed more promising performance.

The reason why the performance of the impact-based weight score function is good is apparent. As can be seen in Eq. (11) and Eq. (12), the BAN has been learned in a way that predicts techniques with high attack impact well due to the design of the score function. Therefore, in the Bayesian network, the relationships between unimportant nodes are weakened, and the relationships between nodes close to the attacker's final goal are strengthened. This way, it is possible to obtain more effective prediction results for blocking an attacker's ultimate goal, which is the ultimate goal of the APT defense system. In conclusion, the experimental results of RQ3 indicate that the BAN is more effective in defending against APT attacks.

The impact-based weight score functions proposed in this paper performed better than the existing score functions. Among them, *wbde* performed the best, at most 25% more.

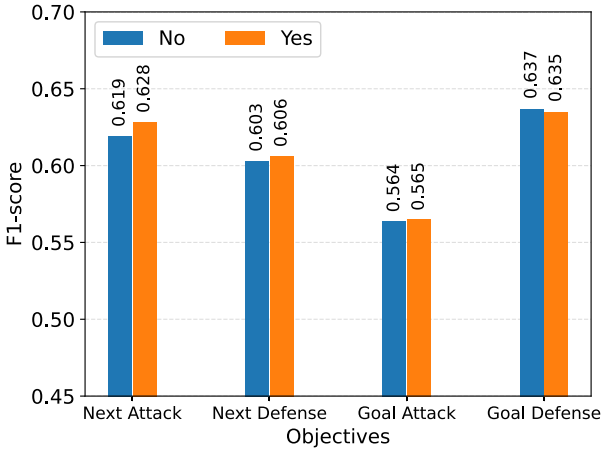

FIGURE 6. Performance comparison of BAN with and without expert knowledge.

4) RQ4. DOES APPLYING EXPERT KNOWLEDGE TO BAN IMPROVE PERFORMANCE?

Also, we experimented to figure out whether the expert knowledge, described in Section V-B, improves the performance of the BAN or not. We transformed the prepared expert knowledge into a blacklist and compared the performance of the BAN with blacklist and without a blacklist. The result is described in Fig. 6.

Consequently, the performance with expert knowledge was usually better, but there was no significant difference. For each objective, The difference in the f1-score of BAN with expert knowledge and without expert knowledge was less than 1.4%. One of the reasons may be that the relationships we identified do not appear in the dataset. As more massive datasets are collected and the diversity of the datasets increases, we expect that there will be a significant increase in the model's performance with our expert knowledge.

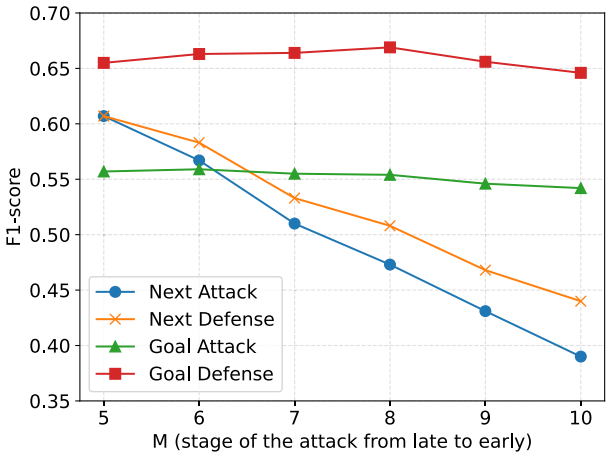

FIGURE 7. Performance comparison at different stages of an attack.

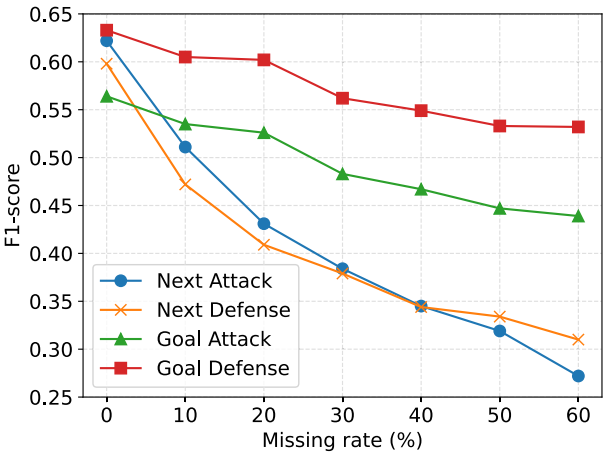

FIGURE 8. Performance comparison across different missing rates.

Even if expert knowledge was utilized in the structure learning, there was no significant performance improvement ($< 1.4\%$).

5) RQ5. HOW IS THE PREDICTION PERFORMANCE DIFFERENT DEPENDING ON THE ATTACK PROGRESS?

We tested how the prediction accuracy changes according to the progress of the attack. The experiment was conducted by predicting the last K attacks when all previous attacks were already detected except the last M attacks. At this time, we sequentially increased the M from 5 to 10 to check out the performance in the early and late stages of the attack. For this experiment, we fixed K to 5. We selected the model that showed the best performance in the previous experiment. The results of RQ5 are shown in Fig. 7.

As a consequence, the performance slightly decreased when the prediction was carried out at the earlier stage of the attack. In the case of M equals 5, the f1-score of the next attack prediction was 0.607, whereas when M equals 10, the f1-score dropped to 0.390. Although the performance of the remaining three objectives decreased also, the degree of performance decline was moderate in predicting the goal attack technique and defense technique. The f1-score of the goal attack prediction gradually decreased from 0.557 when M equals 5 to 0.542 when M equals 10.

Through the results of this experiment, it was found that BAN showed satisfactory performance even in the early and middle stages of the APT attack. Of course, the performance degradation occurred due to the lack of attack information, but it was insignificant. In particular, the performance degradation was minimal when predicting the final attack goal, which is the essential objective of the BAN. In conclusion, BAN can effectively predict attacks and propose defense techniques even in the early stages of an attack.

If the BAN was operated at the early stage of the attack, the prediction performance was decreased ($0.607 \rightarrow 0.390$) However, the performance decrease was slight in the case of the goal attack prediction ($0.557 \rightarrow 0.542$).

6) RQ6. EVEN IF THE PAST ATTACK DETECTION HAS FAILED, CAN OUR MODEL PREDICT THE FUTURE ATTACKS AND DEFENSES ACCURATELY?

Finally, under the assumption that the attack detection system is imperfect, we checked how the performance of the BAN changes depending on detection precision. In the previous experiments, we assumed that 100% of the attack was detected and passed to the BAN. However, there would be attacks that are not detected in the actual detection system. Therefore, to test whether the performance of the BAN is stable even with the inaccurate detection system, we increased the probability of detection failure (missing rate) from 0% to 60% and checked the results. The detection failure probability is the probability that the attack detection will fail. That is, if the detection failure probability equals 10%, only 90% of occurred attacks are detected. Also, the experimented model was the model that showed the best performance in previous research questions. The performance of BAN in an incomplete detection environment is shown in Fig. 8.

Looking at the experimental results, the performance of the BAN declined as the detection failure probability increased. For example, when the attacks were entirely detected, the f1-score of the next attack prediction was 0.628. However, when 60% of attacks were not detected, the f1-score dropped to 0.272. Furthermore, the degree of decline was not significant when predicting the goal attack techniques. In the case of the goal attack prediction objective, the f1-score decreased from 0.564 to 0.439.

This experiment showed that BAN performs well even in an incomplete detection environment. Of course, the prediction performance was insufficient when more than half of the attacks were not detected. However, this is because of the lack of attack information rather than a design problem of BAN. Also, these unreasonable assumptions are unlikely to occur, considering that the accuracy of the detection system, used in the real-world, is mostly over 90%. Under the assumption of a detection accuracy greater than 90%, the performances of

the BAN were above 0.5, with the exception of the prediction of the next defense. Therefore, even if the detection system cannot detect some attacks, BAN can achieve its original objectives.

If the attack detection system is imperfect, the performance of the BAN was degraded by at least 16%. However, it showed acceptable performance (> 0.511) on a realistic attack detection system (missing rate $\leq 10\%$).

VII. CONCLUSION

Although an APT attack has emerged as the first topic of cyber defense, predicting and defending the APT attack has not been studied much. Since the current defense systems can only defend a part of the APT attack, a novel system is required to prevent the attacker's ultimate goal.

In this paper, we proposed the Bayesian ATT&CK Network, a cyberattack prediction model using Bayesian networks. BAN precisely models the attacker's behavior based on MITRE ATT&CK and uses it for attack prediction. Unlike existing studies, it does not depend on raw alerts, so it can be utilized independently for network structures and detection systems. In addition, the model's reliability is improved by learning the model based on the data obtained from the existing APT attack cases. Furthermore, the model's usefulness is maximized by providing not only prediction results but also defense techniques corresponding to the predicted attack. In the performance experiments, BAN showed an f1-score of 0.628 and 0.606, at the next attack prediction and the next defense prediction, respectively. Moreover, in terms of predicting the goal attack and defense, BAN showed an f1-score of 0.565 and 0.634, respectively.

For future research, we will continue to collect additional real-world APT data to improve the performance of our model. Also, modeling and utilizing the additional attack artifacts, such as files, directories, user accounts, or other system activities, can improve the performance. In addition, we plan to create a virtual network that mimics a real-world network for additional experiments in real-world scenarios. Finally, we will study an automated attack response system based on attack prediction results rather than simply suggesting countermeasures.

REFERENCES

[1] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Proc. IFIP Int. Conf. Commun. Multimedia Secur.* Springer, 2014, pp. 63–72.

[2] R. S. Mueller, "Report on the investigation into Russian interference in the 2016 presidential election," US Dept. Justice, Washington, DC, USA, 2019, vol. 1. [Online]. Available: https://www.justice.gov/archives/sco/file/1373816/download

[3] T. Hughes and O. Sheyner, "Attack scenario graphs for computer network threat analysis and prediction," *Complexity*, vol. 9, no. 2, pp. 15–18, Nov. 2003.

[4] A. A. Ramaki, M. Amini, and R. E. Atani, "RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection," *Comput. Secur.*, vol. 49, pp. 206–219, Mar. 2015.

[5] M. Ghasemigol, A. Ghaemi-Bafghi, and H. Takabi, "A comprehensive approach for network attack forecasting," *Comput. Secur.*, vol. 58, pp. 83–105, May 2016.

[6] H. Farhadi, M. AmirHaeri, and M. Khansari, "Alert correlation and prediction using data mining and HMM," *ISC Int. J. Inf. Secur.*, vol. 3, no. 2, pp. 77–101, 2011.

[7] H. A. Kholidy, A. Erradi, S. Abdelwahed, and A. Azab, "A finite state hidden Markov model for predicting multistage attacks in cloud systems," in *Proc. IEEE 12th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2014, pp. 14–19.

[8] A. S. Sendi, M. Dagenais, M. Jabbarifar, and M. Couture, "Real time intrusion prediction based on optimized alerts with hidden Markov model," *J. Netw.*, vol. 7, no. 2, p. 311, Feb. 2012.

[9] A. A. Ramaki, M. Khosravi-Farmad, and A. G. Bafghi, "Real time alert correlation and prediction using Bayesian networks," in *Proc. 12th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2015, pp. 98–103.

[10] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *Proc. 20th Annu. Comput. Secur. Appl. Conf.*, 2004, pp. 370–379.

[11] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Trans. Depend. Sec. Comput.*, vol. 9, no. 1, pp. 61–74, Jan. 2012.

[12] O. B. Fredj, A. Mihoub, M. Krichen, O. Cheikhrouhou, and A. Derhab, "CyberSecurity attack prediction: A deep learning approach," in *Proc. 13th Int. Conf. Secur. Inf. Netw.*, Nov. 2020, pp. 1–6.

[13] T. Li, Y. Jiang, C. Lin, M. S. Obaidat, Y. Shen, and J. Ma, "DeepAG: Attack graph construction and threats prediction with bi-directional deep learning," *IEEE Trans. Depend. Sec. Comput.*, vol. 20, no. 1, pp. 740–757, Jan. 2023.

[14] V. Lisỳ, R. Píbil, J. Stiborek, B. Bošanskỳ, and M. Pěchouček, "Game-theoretic approach to adversarial plan recognition," in *Proc. 20th Eur. Conf. Artif. Intell.*, 2012, pp. 546–551.

[15] R. Píbil et al., "Game theoretic model of strategic honeypot selection in computer networks," in *Decision and Game Theory for Security: Third International Conference, GameSec 2012, Budapest, Hungary, November 5–6, 2012. Proceedings 3*. Berlin, Germany: Springer, 2012.

[16] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. (2018). *MITRE ATT&CK—Design and Philosophy*. [Online]. Available: https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

[17] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proc. Workshop New Secur. Paradigms*, Jan. 1998, pp. 71–79.

[18] S. Bistarelli, F. Fioravanti, and P. Peretti, "Defense trees for economic evaluation of security investments," in *Proc. 1st Int. Conf. Availability, Rel. Secur. (ARES)*, 2006, p. 8.

[19] S. Bistarelli, P. Peretti, and I. Trubitsyna, "Analyzing security scenarios using defence trees and answer set programming," *Electron. Notes Theor. Comput. Sci.*, vol. 197, no. 2, pp. 121–129, Feb. 2008.

[20] A. Roy, D. S. Kim, and K. S. Trivedi, "Cyber security analysis using attack countermeasure trees," in *Proc. 6th Annu. Workshop Cyber Secur. Inf. Intell. Res.*, Apr. 2010, pp. 1–4.

[21] A. Roy, D. S. Kim, and K. S. Trivedi, "Attack countermeasure trees (ACT): Towards unifying the constructs of attack and defense trees," *Secur. Commun. Netw.*, vol. 5, no. 8, pp. 929–943, Aug. 2012.

[22] B. Kordy et al., "Foundations of attack–defense trees," in *Formal Aspects of Security and Trust: 7th International Workshop, FAST 2010, Pisa, Italy, September 16–17, 2010. Revised Selected Papers 7*. Berlin, Germany: Springer, 2011.

[23] B. Kordy, S. Mauw, S. Radomirovic, and P. Schweitzer, "Attack–defense trees," *J. Log. Comput.*, vol. 24, no. 1, pp. 55–87, Feb. 2014.

[24] T. Sommestad, M. Ekstedt, and P. Johnson, "Combining defense graphs and enterprise architecture models for security analysis," in *Proc. 12th Int. IEEE Enterprise Distrib. Object Comput. Conf.*, Sep. 2008, pp. 349–355.

[25] T. Sommestad, M. Ekstedt, and P. Johnson, "Cyber security risks assessment with Bayesian defense graphs and architectural models," in *Proc. 42nd Hawaii Int. Conf. Syst. Sci.*, 2009, pp. 1–10.

[26] T. Sommestad, M. Ekstedt, and P. Johnson, "A probabilistic relational model for security risk analysis," *Comput. Secur.*, vol. 29, no. 6, pp. 659–679, Sep. 2010.

[27] L. Piètre-Cambacédès and M. Bouissou, "Attack and defense modeling with BDMP," in *Computer Network Security: 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2010, St. Petersburg, Russia, September 8–10, 2010. Proceedings 5*. Berlin, Germany: Springer, 2010.

[28] L. Pièrre-Cambacédès and M. Bouissou, "Beyond attack trees: Dynamic security modeling with Boolean logic driven Markov processes (BDMP)," in *Proc. Eur. Dependable Comput. Conf.*, 2010, pp. 199–208.
[29] K. Durkota, V. Lisý, B. Bošanský, C. Kiekintveld, and M. Pěchouček, "Hardening networks against strategic attackers using attack graph games," *Comput. Secur.*, vol. 87, Nov. 2019, Art. no. 101578.
[30] L.-X. Yang, P. Li, Y. Zhang, X. Yang, Y. Xiang, and W. Zhou, "Effective repair strategy against advanced persistent threat: A differential game approach," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1713–1728, Jul. 2019.
[31] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, and J. Disso, "Cyber-attack modeling analysis techniques: An overview," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud Workshops (FiCloudW)*, Aug. 2016, pp. 69–76.
[32] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti, "A review of threat modelling approaches for APT-style attacks," *Heliyon*, vol. 7, no. 1, Jan. 2021, Art. no. e05969.
[33] M. Lehto, "APT cyber-attack modelling: Building a general model," in *Proc. 17th Int. Conf. Cyber Warfare Secur.*, vol. 17. New York, NY, USA: Academic, 2022, pp. 121–129.
[34] *Cyber Kill Chain | Lockheed Martin*. Accessed: Oct. 27, 2022. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[35] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues Inf. Warfare Secur. Res.*, vol. 1, no. 1, p. 80, 2011.
[36] *MITRE ATT&CK*. Accessed: Oct. 27, 2022. [Online]. Available: https://attack.mitre.org/
[37] B. E. Strom et al., "Finding cyber threats with ATT&CK-based analytics," MITRE Corp., Bedford, MA, USA, Tech. Rep. MTR170202, 2017.
[38] C. Johnson, L. Badger, D. Waltermire, J. Snyder, and C. Skorupka, "Guide to cyber threat information sharing," Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-150, 2016.
[39] J. Basra and T. Kaushik, "MITRE ATT&CK as a framework for cloud threat investigation," Berkeley Center Long-Term Cybersecur., Berkeley, CA, USA, White Paper, 2020. [Online]. Available: https://cltc.berkeley.edu/publication/mitre-attck/
[40] *Contribute | MITRE ATT&CK*. Accessed: Aug. 4, 2023. [Online]. Available: https://attack.mitre.org/resources/contribute/
[41] *Updates | MITRE ATT&CK*. Accessed: Aug. 4, 2023. [Online]. Available: https://attack.mitre.org/resources/updates/
[42] Joint Task Force Transformation Initiative, "Security and privacy controls for federal information systems and organizations," Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, NIST SP 800-53, 2013, pp. 8–13.
[43] Center of Internet Security. (2021). *CIS Controls Version 8*. [Online]. Available: https://www.cisecurity.org/controls/v8/
[44] *Information Technology—Security Techniques—Information Security Management Systems—Overview and Vocabulary*, ISO/IEC, Geneva, Switzerland, 2018.
[45] Microsoft. *Overview of the Azure Security Benchmark V2 | Microsoft Docs*. Accessed: Oct. 27, 2022. [Online]. Available: https://docs.microsoft.com/en-us/security/benchmark/azure/overview
[46] *Mitigations—Enterprise | MITRE ATT&CK*. Accessed: Oct. 27, 2022. [Online]. Available: https://attack.mitre.org/mitigations/enterprise/
[47] C. Fowler, M. Goffin, B. Hill, R. Lamourine, and A. Sovern, "An introduction to MITRE shield," MITRE Corp., McLean, VA, USA, Tech. Rep., 2020. [Online]. Available: https://shield.mitre.org/resources/downloads/Introduction_to_MITRE_Shield.pdf
[48] *Engage Home*. Accessed: Oct. 27, 2022. [Online]. Available: https://engage.mitre.org/
[49] *A Practical Guide to Adversary Engagement*, MITRE Corp., McLean, VA, USA, 2022.
[50] *D3FEND Matrix | MITRE D3FENDT*. Accessed: Oct. 27, 2022. [Online]. Available: https://d3fend.mitre.org/
[51] P. E. Kaloroumakis and M. J. Smith, "Toward a knowledge graph of cybersecurity countermeasures," MITRE Corp., McLean, VA, USA, Tech. Rep. PRS-20-2034, 2021. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD1156977.pdf
[52] L. M. De Campos and N. Friedman, "A scoring function for learning Bayesian networks based on mutual information and conditional independence tests," *J. Mach. Learn. Res.*, vol. 7, no. 10, pp. 2149–2187, 2006.
[53] D. M. Chickering, "A transformational characterization of equivalent Bayesian network structures," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, 1995, pp. 87–98.
[54] T. Silander, T. Roos, P. Kontkanen, and P. Myllymäki, "Factorized normalized maximum likelihood criterion for learning Bayesian network structures," in *Proc. 4th Eur. Workshop Probabilistic Graph. Models (PGM)*, 2008, pp. 257–272.
[55] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, Sep. 1995.
[56] W. Buntine, "Theory refinement on Bayesian networks," in *Uncertainty in Artificial Intelligence*. Amsterdam, The Netherlands: Elsevier, 2014, p. 52.
[57] M. Scutari, "An empirical-Bayes score for discrete Bayesian networks," in *Proc. Conf. Probabilistic Graph. Models*, 2016, pp. 438–448.
[58] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, Oct. 1992.
[59] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010.
[60] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.
[61] P. Larranaga, M. Poza, Y. Yurramendi, R. H. Murga, and C. M. H. Kuijpers, "Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 912–926, Sep. 1996.
[62] S. Lee and S. B. Kim, "Parallel simulated annealing with a greedy algorithm for Bayesian network structure learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1157–1166, Jun. 2020.
[63] N. Friedman and D. Koller, "Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks," *Mach. Learn.*, vol. 50, nos. 1–2, pp. 95–125, 2003.
[64] D. Colombo and M. H. Maathuis, "Order-independent constraint-based causal structure learning," *J. Mach. Learn. Res.*, vol. 15, no. 116, pp. 3921–3962, 2014.
[65] D. Margaritis, "Learning Bayesian network model structure from data," Ph.D. dissertation, U.S. Army, Arlington County, VA, USA, 2003.
[66] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, Oct. 2006.
[67] N. Friedman, I. Nachman, and D. Peér, "Learning Bayesian network structure from massive datasets: The 'sparse candidate' algorithm," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, 1999, pp. 206–215.
[68] D. J. Spiegelhalter and S. L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, no. 5, pp. 579–605, Aug. 1990.
[69] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*, vol. 405. Hoboken, NJ, USA: Wiley, 2009.
[70] R. Chang and W. Wang, "Novel algorithm for Bayesian network parameter learning with informative prior constraints," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
[71] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach," in *Proc. 2nd AAAI Conference Artif. Intell. (AAAI)*. Pittsburgh, PA, USA: AAAI Press, 1982, pp. 133–136.
[72] N. L. Zhang and D. Poole, "A simple approach to Bayesian network computations," in *Proc. 10th Can. Conf. Artif. Intell.*, 1994, pp. 1–8.
[73] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *Int. J. Approx. Reasoning*, vol. 15, no. 3, pp. 225–263, Oct. 1996.
[74] A. F. M. Smith and G. O. Roberts, "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods," *J. Roy. Stat. Soc. B, Methodol.*, vol. 55, no. 1, pp. 3–23, Sep. 1993.
[75] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.
[76] A. Althnian, D. AlSaeed, H. Al-Baity, A. Samha, A. B. Dris, N. Alzakari, A. A. Elwafa, and H. Kurdi, "Impact of dataset size on classification performance: An empirical evaluation in the medical domain," *Appl. Sci.*, vol. 11, no. 2, p. 796, Jan. 2021.
[77] A. Oniśko, M. J. Druzdzel, and H. Wasyluk, "Learning Bayesian network parameters from small data sets: Application of noisy-OR gates," *Int. J. Approx. Reasoning*, vol. 27, no. 2, pp. 165–182, Aug. 2001.
[78] A. M. MacAllister, "Investigating the use of Bayesian networks for small dataset problems," Ph.D. dissertation, Dept. Mech. Eng., Iowa State Univ., Ames, IA, USA, 2018.

[79] Y. Hou, E. Zheng, W. Guo, Q. Xiao, and Z. Xu, "Learning Bayesian network parameters with small data set: A parameter extension under constraints method," *IEEE Access*, vol. 8, pp. 24979–24989, 2020.
[80] S. Gao and X. Wang, "Quantitative utilization of prior biological knowledge in the Bayesian network modeling of gene expression data," *BMC Bioinf.*, vol. 12, no. 1, pp. 1–13, Dec. 2011.
[81] R. Al-Shaer, J. M. Spring, and E. Christou, "Learning the associations of MITRE ATT & CK adversarial techniques," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2020, pp. 1–9.
[82] M. Zissman, "DARPA intrusion detection scenario specific data sets," Tech. Rep., 2000.
[83] S. Myneni, A. Chowdhary, A. Sabur, S. Sengupta, G. Agrawal, D. Huang, and M. Kang, "DAPT 2020-constructing a benchmark dataset for advanced persistent threats," in *Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020, Proceedings 1*. Springer, 2020, pp. 138–163.
[84] S. Myneni, K. Jha, A. Sabur, G. Agrawal, Y. Deng, A. Chowdhary, and D. Huang, "Unraveled—A semi-synthetic dataset for advanced persistent threats," *Comput. Netw.*, vol. 227, May 2023, Art. no. 109688.
[85] *APT & CyberCriminal Campaign Collection*. Accessed: Oct. 27, 2022. [Online]. Available: https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections
[86] *Best Practices for MITRE ATT&CK Mapping*, Cybersec. Infrastruct. Secur. Agency, Arlington, VA, USA, 2021.
[87] *Module 2: Mapping to ATT&CK From a Finished Report*, Training Resour., MITRE Corp., McLean, VA, USA, 2019.
[88] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated retrieval of ATT&CK tactics and techniques for cyber threat reports," in *Proc. 1st Cyber Threat Intell. Symp. (CTI)*, Mar. 2020. [Online]. Available: https://www.first.org/events/symposium/zurich2020/
[89] V. S. M. Legoy, "Retrieving ATT&CK tactics and techniques in cyber threat reports," M.S. thesis, Univ. Twente, Enschede, The Netherlands, 2019.
[90] *Threat Report ATT&CK Mapper—TRAM—CTID*. Accessed: Oct. 27, 2022. [Online]. Available: https://ctid.mitre-engenuity.org/our-work/tram/
[91] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with Bayesian networks: A bootstrap approach," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, 1999, pp. 196–205.
[92] B. M. Broom, K.-A. Do, and D. Subramanian, "Model averaging strategies for structure learning in Bayesian networks with limited data," *BMC Bioinf.*, vol. 13, no. S13, pp. 1–18, Aug. 2012.
[93] J. S. Ide and F. G. Cozman, "Random generation of Bayesian networks," in *Advances in Artificial Intelligence: 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002, Porto de Galinhas/Recife, Brazil, November 11–14, 2002, Proceedings*, vol. 2507. Berlin, Germany: Springer, 2003.
[94] A. C. Constantinou, Z. Guo, and N. K. Kitson, "The impact of prior knowledge on causal structure learning," *Knowl. Inf. Syst.*, vol. 65, pp. 3385–3434, Apr. 2023.
[95] R. D. Shachter and M. A. Peot, "Simulation approaches to general probabilistic inference on belief networks," in *Proc. 5th Annu. Conf. Uncertainty Artif. Intell.*, 1990, pp. 221–234.
[96] M. Scutari, "Learning Bayesian networks with the bnlearn R package," *J. Stat. Softw.*, vol. 35, no. 3, pp. 1–22, 2010.
[97] P. Suter, J. Kuipers, G. Moffa, and N. Beerenwinkel, "Bayesian structure learning and sampling of Bayesian networks with the R package BiDAG," 2021, *arXiv:2105.00488*.
[98] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Dept. Comput. Sci., Oregon State Univ., Corvallis, OR, USA, Tech. Rep., 2010, p. 25, vol. 18, no. 1.

YOUNGJOON KIM received the B.S. degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2017, where he is currently pursuing the Ph.D. degree in cyber security. He is also a Cybersecurity Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include fuzzing, machine learning for cybersecurity, and Bayesian statistic.

INSUP LEE (Student Member, IEEE) received the B.S. degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing the Ph.D. degree in cybersecurity. He is also a Cybersecurity Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include deep learning, intelligent networks, generative adversarial networks, and AI-based cybersecurity.

HYUK KWON received the B.S. degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing the Ph.D. degree in information security. He is also a Cybersecurity Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include software vulnerability analysis, binary exploitation, data engineering, and cyber-physical security.

KYEONGSIK LEE received the B.E. degree in computer science and engineering from Sejong University, Seoul, Republic of Korea, in 2009, and the M.S. degree in information management and security from Korea University, Seoul, in 2011. He is currently a Senior Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include digital forensics, incident response, and malware analysis.

JIWON YOON received the B.S. degree in information engineering from Sungkyunkwan University, South Korea, in 2003, the M.S. degree in informatics from The University of Edinburgh, U.K., in 2004, and the Ph.D. degree in statistical signal processing from the University of Cambridge, U.K., in 2008.

From 2008 to 2009, he was a Postdoctoral Research Assistant with the Robotics Group, University of Oxford, U.K. From 2009 to 2011, he was a Research Fellow with the Statistics Department, Trinity College Dublin, Ireland. He was a Research Scientist with the IBM Research Laboratory, from 2011 to 2012. From 2012 to 2016, he was an Assistant Professor with the Cyber Defense Department, Korea University, where he has been a Professor with the School of Cybersecurity, since 2021. His research interests include intelligence, such as signal intelligence, crypto intelligence, artificial intelligence, and open source intelligence.