symmetry

Article

Missing Data Imputation Based on Causal Inference to Enhance Advanced Persistent Threat Attack Prediction

Xiang Cheng [1,2,*], Miaomiao Kuang [1] and Hongyu Yang [3]

[1] School of Information Engineering, Yangzhou University, Yangzhou 225127, China; 17772229647@163.com
[2] Key Laboratory of Flying Internet, Civil Aviation University of China, Tianjin 300300, China
[3] School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China
* Correspondence: huozhai9527@126.com

Abstract: With the continuous development of network security situations, the types of attacks increase sharply, but can be divided into symmetric attacks and asymmetric attacks. Symmetric attacks such as phishing and DDoS attacks exploit fixed patterns, resulting in system crashes and data breaches that cause losses to businesses. Asymmetric attacks such as Advanced Persistent Threat (APT), a highly sophisticated and organized form of cyber attack, because of its concealment and complexity, realize data theft through long-term latency and pose a greater threat to organization security. In addition, there are challenges in the processing of missing data, especially in the application of symmetric and asymmetric data filling, the former is simple but not flexible, and the latter is complex and more suitable for highly complex attack scenarios. Since asymmetric attack research is particularly important, this paper proposes a method that combines causal discovery with graph autoencoder to solve missing data, classify potentially malicious nodes, and reveal causal relationships. The core is to use graphic autoencoders to learn the underlying causal structure of APT attacks, with a special focus on the complex causal relationships in asymmetric attacks. This causal knowledge is then applied to enhance the robustness of the model by compensating for data gaps. In the final phase, it also reveals causality, predicts and classifies potential APT attack nodes, and provides a comprehensive framework that not only predicts potential threats, but also provides insight into the logical sequence of the attacker's actions.

Keywords: causal discovery; APT attack; attack predict; graph autoencoder

check for updates

Citation: Cheng, X.; Kuang, M.; Yang, H. Missing Data Imputation Based on Causal Inference to Enhance Advanced Persistent Threat Attack Prediction. *Symmetry* **2024**, *16*, 1551. https://doi.org/10.3390/sym16111551

Academic Editor: Zhixun Su

Received: 22 October 2024
Revised: 15 November 2024
Accepted: 15 November 2024
Published: 19 November 2024

1. Introduction

In the current network security environment, symmetric attack usually refers to the power parity between the attacker and the defender, with both sides being relatively balanced in terms of technology and resources. These attacks often manifest as direct technical confrontations, where the attacker launches raids or exploits vulnerabilities for direct assaults. Such attacks are usually easy to detect and have a short impact time duration, but they can quickly paralyze the system or cause data loss [1]. Most of these attacks are common means of attacking, and the countermeasures and solutions are relatively mature. On the other hand, asymmetric attacks are attacks characterized by a significant imbalance between the attacker and the defender [2,3]. Taking typical APT attacks as an example, their attacks are usually launched by experienced attack organizations, often with strong concealment, and achieve the purpose of attack through long-term latency, slow penetration, and data collection [4]. One characteristic that distinguishes APT attacks from other attacks is its multi-stage attack strategy, from initial penetration, lateral movement to later data extraction, the attacker may adjust the means at each stage to bypass the defense. If a successful attack is carried out, the damage will be enormous. For this type of attack, the multi-stage attack strategy [5] and complexity [6] will be a major challenge in the research.

Multi-stage attacks. In the field of cybersecurity, the detection and analysis of multi-stage APT attacks has become a critical area of recent research. Neuschmied et al. [7] utilized multi-stage autoencoders for APT attack detection, employing deep learning to capture the nuanced features of attack behaviors, enhancing detection accuracy. Wilkens et al. [8] constructed Kill Chain State Machines to track and link alerts across different stages, thereby building potential multi-stage attack scenarios. Xie et al. [5] proposed a multi-stage APT attack detection method based on sample enhancement, combining Transformer encoders with a multi-stage awareness mechanism to improve the selection of normal traffic. Zhou et al. [9] detected multi-stage attacks using a sequence-to-sequence model, particularly suited for APT attacks occurring over extended periods. Takey et al. [10] presented a real-time early detection method for multi-stage attacks, focusing on network traffic to swiftly identify and respond to multi-stage intrusions. These studies not only enrich the theoretical and practical aspects of APT attack detection but also provide robust technical support for network security defenses. While existing methods offer substantial advancements in APT detection, the integration of causal analysis promises to further sharpen predictive accuracy and defensive capabilities.

Complexity. Graphs provide an intuitive way to represent complex relationships and patterns, and the combination of causality and graphs can more intuitively analyze apt attacks. Like a framework that extends Bayesian causal discovery [11], causal relationships are modeled by sampling the posterior distribution in the directed acyclic graph (DAG) space. Additionally, the ability of graph neural networks to generalize beyond the distribution graph of the training data has also been explored, showing the potential of causal analysis in ensuring model consistency [12]. Explorations into nonlinear causal discovery, particularly with latent confounders, open avenues for uncovering obscured causal relationships, surpassing conventional detection limits [13]. Causality-based neural network repair methods underscore the importance of causal insights in model correction and reliability [14]. Moreover, employing causal correlation with semantic analysis for multi-stage threat detection showcases the utility of causal approaches in identifying sophisticated cyber threats [15].

In this paper, we develop a causal prediction model for APT attacks, introducing an innovative approach to more accurately understand and predict APT attack behavior and outcomes. Using data from the DARPA TC dataset [16], which provides representative samples of attack behavior within complex network environments, we apply the LADIES sampling method to efficiently extract representative samples from large-scale datasets. This method helps to reduce data processing demands while preserving key attack characteristics. The causal structure of APT attack is learned by autoencoders and the potential causal relationship is revealed. The causal structure is used to make up the missing data and enhance the robustness of the model. Finally, the causal relationship is revealed to predict the nodes of APT attacks. The innovation points of this paper are as follows:

- The combination of causal discovery and APT attack motivation analysis reveals event correlation through causality and infers potential attack strategies and motivations.
- The causal driven data imputation method is used to ensure that the imputation results are closer to the real data and better reflect the causal mechanism in the data, which is more accurate than traditional imputation methods.
- The causal mechanism is integrated into the generative model, which is not only used for data reconstruction, but also can mine the potential causal chain, so as to learn the causal dependence in the latent space more accurately.

2. Related Work

In this section, we will explore the existing limitations of APT attack detection and explore what needs to be done from those limitations.

The current detection methods for APT attacks still have some limitations, including the problem of data sparsity [17]. No matter what kind of system is used for data acquisition, a certain degree of data loss will inevitably occur in the process of data acquisition, resulting in data loss [18]. This makes it difficult to comprehensively track the attack behavior. Traditional APT attack detection methods rely on the integrity of the data, but the missing data will lead to the deviation of the behavior pattern analysis, thus missing the key attack clues. The main existing strategies for solving missing data include linear-based approaches, interpolation-based approaches, model-based approaches, and deep learning-based approaches.

The two basic interpolation methods, linear interpolation [19] and mean interpolation, fail to capture complex patterns and trends in the data, and neither takes causality into account. The regression-based multiple imputation method [20] is a common statistical technique for managing missing data, which uses regression models to predict missing values. Although the effect is best at a particular loss rate, as the loss rate changes, extremely unstable problems also come out. Model-based approaches [21] illustrate the application of these data in different statistical models to address data missing problems. For example, maximum likelihood estimation (MLE) is used, but by comparison, his performance is mediocre and cannot effectively deal with the nonlinear characteristics of the data. Based on deep learning technologies such as generative adversarial model GAIN [22], which can predict missing values by generator and distinguish filling effects by discriminator. In the experimental part, the method of deep learning is superior to other methods, so the part of filling missing data is also solved by the deep learning method. Several existing techniques are evaluated in the selection process. The traditional deep Convolutional Neural Network (CNN) is very effective in local feature extraction [23], but it mainly focuses on local information; when dealing with APT attacks, a single feature extraction method cannot fully capture complex causal relationships [22]. Therefore, when CNN is used alone, it is difficult to fully capture the multi-level causal relationships in the face of complex APT attack chains. Similarly, time series models such as recurrent neural networks (RNN) and long short-term memory networks (LSTM) perform well in processing continuous time data and capturing time dependence [24], and they can cope with time-related features in APT attacks to a certain extent. However, these models mainly focus on temporal dependence and have limited effectiveness in dealing with multi-level causality and cross-domain dependency of APT attacks. In addition, RNNS and LSTMS often rely on interpolation or simple filling strategies when faced with missing data, which makes them perform poorly on high-dimensional sparse datasets, especially when missing values in the data carry important causal information. This makes them perform poorly on high-dimensional sparse datasets, especially when missing values in the data carry important causal information. In contrast, the graph autoencoder (GAE) is better able to model complex cause-and-effect relationships and dependencies between events by learning directly on the graph structure [25]. GAE with CNN as encoder and LSTM as decoder, it overcomes the limitations of previous interpolation methods that rely on oversimplified assumptions and invalidate the actual data, and also avoids the drawbacks of model-based methods that may select inappropriate models and lead to biased results. And GAE adds a causal drive to make the data more causal. This is a cause-driven interpolation strategy [26], which uses the causal mechanisms implicit in the data to guide the filling of missing data.

Another significant challenge is the identification of complex attack chains [27]. APT attacks often consist of a series of carefully planned and hidden steps, from reconnaissance, initial intrusion, lateral movement, and data penetration, forming a complex attack path. Traditional detection methods rely on predefined rules or pattern matching, and it is difficult to effectively identify the complete attack chain when facing the flexible tactics of attackers. Especially when there is no obvious connection between different events, these methods tend to overlook important links in the attack chain. In order to overcome this limitation, the model classifies the event types, distinguishes the events of different attack stages (such as initial intrusion, lateral movement, etc.), and deeply analyzes the characteristics of each type of event. Through this classification analysis, the model can more accurately capture the correlation between different stages, so as to reconstruct the complete attack chain. Combined with causal reasoning techniques, the model can also reveal implicit causal associations across events, help identify key causal links, and make the attack path clearer. This not only improves the accuracy of APT detection, but also enhances the model's ability to interpret complex attack chains.

3. Materials and Methods

This scenario uses host-based intrusion detection: the attacker illegally infiltrates the host, gains system privileges, and steals important information from the host by improper means. The stages of an APT attack are typically divided into initial breach, command and control (C&C) system setup, lateral movement, and data exfiltration [8], with various techniques used at each stage. Assume that log information about apt attacks has been collected from the host before stealing important information. The overall framework includes three parts: subgraph sampling, autoencoder interpolation of missing data and causal prediction, as shown in Figure 1.

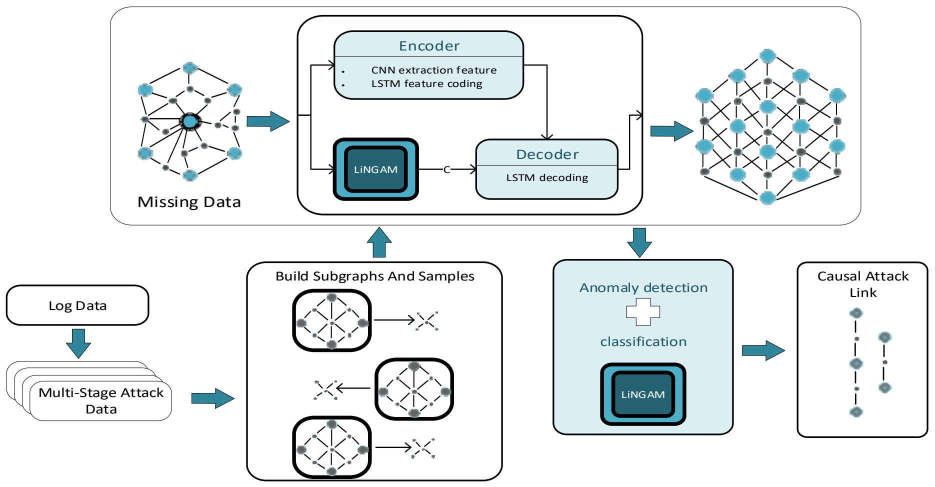

Figure 1. System architecture.

Initially, the log data are converted into graph data for sampling analysis, and the causal matrix is used to guide the graph autoencoder to fill in the missing data. This ensures that the model not only achieves high accuracy in data reconstruction and interpolation but also preserves causal relationships within the data. Finally, potential malicious nodes in the dataset are predicted and classified by combining causal inference with malicious node detection, thereby revealing their causal relationships. Each module of the system is discussed in detail in the following sections.

3.1. Pre-Processing and Sampling

In the data pre-processing stage, we apply a network security data processing method based on Avro and APT attack stages. First, we unpack and read the raw binary, then parse the data with the full Avro schema to extract the important event records. To deal with non-JSON serializable objects, a conversion function is designed to convert complex data structures into serializable formats and save them as JSON files. Next, each stage of APT attack life cycle is defined. The stages of an APT attack tend to stagger and may cycle back and forth. For example, reconnaissance occurs repeatedly in each stage, lateral movement and persistence are a continuous process, and exfiltration can occur simultaneously at any stage rather than the last step.

Including reconnaissance, intrusion, lateral movement, persistence, execution, and exfiltration, our definition correlates the relevant event types for each phase and iterates through the data to save the classification of events for each phase into a separate JSON file. Table 1 below shows the details of the event.

Table 1. Details of event.

Data Tpye	Description
EVENT_CONNECT	establishes a network connection
EVENT_READ	reads data
EVENT_RECVFROM	receives data from a network connection
EVENT_EXECUTE	executes a program or command
EVENT_OPEN	opens a file or resource
EVENT_FORK	creates a new process
EVENT_WRITE	writes data to a file or output stream
EVENT_CREATE_OBJECT	creates an object, such as a file or process
EVENT_SENDMSG	sends the message
EVENT_CLOSE	closes a file or network connection
EVENT_SENDTO	sends data to a network connection

The original JSON format data are converted into graph data, followed by feature encoding and sampling processing to generate data suitable for graph neural network modeling. Through a series of data processing, a directed graph containing nodes and edges is constructed, node features are encoded and standardized, and subgraphs are extracted from each graph using LADIES sampling method. Additionally, the collected dataset is categorized by APT attack type, and the entire dataset is further segmented according to the attack life cycle, as detailed in the following Table 2.

Table 2. Different stages of events.

Stage	Data Type		
Reconnaissance	EVENT_CONNECT	EVENT_READ	EVENT_RECVFROM
Intrusion	EVENT_EXECUTE	EVENT_OPEN	
Lateral Movement	EVENT_FORK	EVENT_CONNECT	EVENT_EXECUTE
Persistence	EVENT_WRITE	EVENT_CREATE_OGJECT	
Execution	EVENT_WRITE	EVENT_SENDMSG	EVENT_CLOSE
Exfiltration	EVENT_SENDTO	EVENT_WRITE	EVENT_CLOSE

To efficiently process and analyze the data, we begin loading the APT attack phase data and storing it in a list. As each file of event data is read, each record is treated as an individual event, containing fields like the unique identifier uuid, subject, predicateObject, and other relevant details. Each event is represented as a node of a graph, with directed edges created between the nodes based on the subject and predicateObject fields, thus forming a directed graph structure. We iterate through the nodes in the graph, determine the set of attributes that all nodes should have, and fill in the default values for the nodes that lack attributes to ensure the consistency of the graph data. We then convert all node property values to a string format for subsequent tag encoding. We construct a label encoder for each attribute, map the attribute values to a numerical representation, and generate a node feature matrix and scale the features using a standardized method so that their mean is 0 and their standard deviation is 1. Assume that the original data matrix is X, and its normalization is shown in Formula (1).

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{X} - \mu_X}{\sigma_X} \tag{1}$$

Then, the LADIES [26], Layer-wise Adaptive Node Importance Sampling method was used to sample the graph data. The LADIES algorithm extracts a representative subset of nodes from large-scale graphs through layer-by-layer adaptive importance sampling for subsequent training and analysis of graph neural networks. The specific steps are as follows: first, a certain number of nodes are randomly selected from all nodes as the initial sampling nodes; then, for each sampling node, according to the importance of its neighbor nodes, a fixed number of neighbor nodes are selected to constitute the sampling batch. Next, build a subgraph that contains these sample nodes and their neighbors. Finally, repeat the above steps to expand the sampling nodes layer by layer until the preset number of sampling layers is reached. The advantage of this algorithm is that it can select important adjacent nodes adaptively to ensure the representativeness of the image and has high efficiency in processing large-scale graph data. The algorithm can effectively extract representative subgraphs, and the sampled subgraphs are stored as batch data for subsequent processing.

3.2. Missing Data Processing Based on LiNGAM Graph Autoencoder

This paper introduces a deep learning framework that integrates causal discovery with a graph autoencoder (GAE) to effectively handle complex graph-structured data with missing data. Firstly, the Linear Non-Gaussian Acyclic Model (LiNGAM) [28] algorithm is employed to find the causal relationships within the data, and then these causality relationships are integrated into the training process of the graph autoencoder to achieve a more accurate interpolation of missing data.

3.2.1. Graph Autoencoder

GAE is a core model for dealing with graph-structured data, especially for dealing with missing data. It mainly consists of two parts: encoder and decoder, which are responsible for compressing the input data into a compact representation and reconstructing the original data from it, preserving important structures and causal relationships in the data. The model design is described in Figure 2 below.

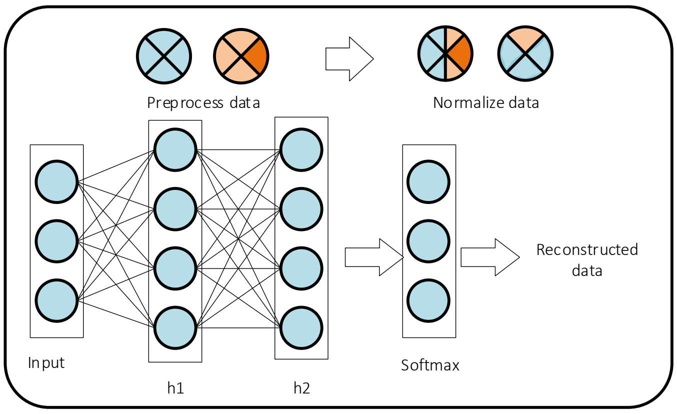

Figure 2. This is a model figure.

The task of the encoder is to map the input data to a low-dimensional latent representation. Firstly, the local features of the input data were extracted by Convolutional Neural Network (CNN). In this process, a one-dimensional convolutional layer (Conv1D) processes graph node features and traverses the input sequence through a sliding window to extract local patterns and relationships at each position. Then, Batch Normalization is performed on the convolution results to keep the distribution of data in each mini-batch consistent and ensure the stability of the network during training. Then, the ReLU activation function is applied to introduce nonlinear characteristics, which helps the model to recognize more complex features and patterns. The observation data matrix is $\mathbf{X} \in \mathbb{R}^{n\times d}$

$$\mathbf{H}_1 = \text{ReLU}(\text{BatchNorm}(\mathbf{X} \cdot \mathbf{W}_{conv} + \mathbf{b}_{conv})) \tag{2}$$

Next, the local features extracted by CNN are passed to the long short-term memory (LSTM) network to further model the time series relationship or dependency between nodes.

Through its internal memory units, the LSTM networks are able to preserve long-term dependency information in the input data and generate compact hidden state representations.

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{H}_1, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \tag{3}$$

In addition, the encoder also maps the hidden states into a "latent causal matrix A" through a fully connected layer. This matrix is used to capture potential causal relationships between the nodes of the graph, identifying which nodes are likely to affect other nodes. In order to ensure the rationality of the causal relationship, the encoder ensures that the causal matrix is acyclic (that is, the causal relationship does not form a closed loop) by the matrix index calculation method.

$$\mathbf{A} = \text{reshape}(\mathbf{W}_\text{A}\text{h}+\mathbf{b}_\text{A}, (d, d)) \tag{4}$$

The decoder's task is to reconstruct the original graph node features from the low-dimensional representation generated by the encoder. The LSTM layer in the decoder receives the hidden state output from the encoder and, by gradually unfolding the information, generates a feature sequence similar to the input data. In this process, the decoder also refers to the causal matrix generated by the encoder to adjust the feature values according to the causal relationship, ensuring that the generated features are not only close to the original data numerically, but also conform to the causal relationship of the data in the logical structure.

$$\mathbf{H}_2 = \text{LSTM}(\mathbf{h}) \tag{5}$$

$$\mathbf{X}_{recon} = \text{FC}(\mathbf{H}_2) \tag{6}$$

The output of GAE is the following two parts: the reconstructed eigenmatrix $\mathbf{X}_\text{recon}$ and the learned potential causal matrix **A**. The reconstruction matrix is compared with the input data to measure the reconstruction error of the model, while the causation matrix is compared with the causation matrix generated by LiNGAM to ensure that the learned causation is reasonable.

3.2.2. LiNGAM

An important innovation of the model is the introduction of causality. In order to consider the causal structure in the learning process, we use the Linear Non-Gaussian Acyclic Model (LiNGAM) algorithm to discover the causal relationship in the data and use the results to guide the training and interpolation process of GAE. The LiNGAM is able to infer causality from data based on a linear non-Gaussian model, which allows the model to not only capture statistical dependencies, but also identify and exploit causal structures in the data. By assuming that the data generation process can be represented as a directed acyclic graph (DAG), each variable is generated by a linear combination of its parent (causal precursor node) with a non-Gaussian noise term. After normalizing the data, we feed it into the LiNGAM algorithm for causal discovery. By analyzing the data, LiNGAM generates a causal matrix **C** that represents the causal relationships between the variables.

3.2.3. Loss Function

In the course of model training, we design a comprehensive loss function to optimize the performance of the model. The core of the loss function is the reconstruction loss, which measures the reconstruction accuracy of the model by calculating the mean square error (MSE) between the reconstructed data of the model and the original input data.

$$\mathcal{L}_{recon} = \frac{1}{N}\sum_{i=1}^{N} ||\hat{X}_{recon,i} - \hat{X}_i||_2^2 \tag{7}$$

In order to ensure that the causal matrix generated by the model is acyclic, the loss function contains acyclic constraints. Through matrix trace operation, the potential loop degree in the causal matrix is calculated, and the loss of this part is minimized to ensure that the generated causal graph is acyclic.

$$L_{\text{acyclic}} = \text{tr}(\text{expm}(\mathbf{A} \odot \mathbf{A})) - d \tag{8}$$

The regularization term is introduced into the loss function to control the complexity of the model and to prevent the model from overfitting. L1 regularization helps to generate sparse causal matrix, that is, to reduce unnecessary causal connections between features. L2 regularization controls the size of matrix elements to avoid excessive model complexity and enhance its generalization ability.

$$\mathcal{L}_{\text{regularization}} = \beta \left|\left|\mathbf{A}\right|\right|_1 + \gamma \left|\left|\mathbf{A}\right|\right|_2^2 \tag{9}$$

The causal constraints are designed to ensure that the causal matrix learned by the model is consistent with the a priori causal matrix discovered by the LiNGAM algorithm. To achieve this, we add a term to the loss function that measures the difference between the two matrices in terms of the L1 norm. Reducing this difference helps the model to reconstruct the data more in line with the known causal structure, which improves the rationality of the interpolation results.

$$\mathcal{L}_{\text{causality}} = \lambda_c \left|\left|\mathbf{A} - \mathbf{C}\right|\right|_1 \tag{10}$$

The final loss function combines the reconstruction error, acyclic constraint, regularization term, and causality constraint. The model continuously optimizes its parameters by minimizing the comprehensive loss function during training.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \alpha \mathcal{L}_{\text{acylic}} + \mathcal{L}_{\text{regularization}} + \mathcal{L}_{\text{causality}} \tag{11}$$

For model optimization, we used the AdamW optimizer, which combines an adaptive learning rate and weight decay to dynamically adjust the learning rate and prevent the model from overfitting. The learning rate scheduler automatically decreases the learning rate when the loss no longer decreases significantly to prevent the model from getting stuck in a local optimum. In addition, an early stopping mechanism is added to the training process, which automatically stops the training when the model performance no longer improves significantly after several iterations to avoid overfitting. After training, the GAE model is used to interpolate the missing data. The encoder and decoder are able to fill in the missing parts of the data reasonably and recover the original scale of the data by de-normalization. The accuracy of the interpolated data is assessed by means of mean square error (MSE) and mean absolute error (MAE), thus providing a quantitative evaluation of the model interpolation performance.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left|\left|\hat{X}_{\text{filled},i} - X_i\right|\right|_2^2 \tag{12}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left|\left|\hat{X}_{\text{filled},i} - X_i\right|\right|_1 \tag{13}$$

3.3. Predictions

We combine the dual methods of causal inference and malicious node detection to effectively identify potential malicious nodes in the dataset and reveal their causal relationship. Firstly, we used the Linear Non-Gaussian Acyclic Model to conduct causality analysis on standardized data, extract the causal chain between variables, and predict its impact on other variables through the intervention of specific variables. Next, an isolated

Forest algorithm is used for anomaly detection. Isolated forest is an unsupervised learning method, based on the idea of random forests, specifically designed to find anomalies in data. The algorithm constructs multiple trees by randomly selecting features and segmentation points and uses the structure of the tree to isolate data points. Outliers are generally easier to isolate and therefore have a shorter path through the tree. In this approach, isolated forests are used to detect potentially malicious nodes, which are flagged as anomalies for subsequent classification model processing.

To further classify and confirm these abnormal nodes labeled by isolated forests, a deep neural network (DNN) model is designed and implemented. Unlike traditional fully connected networks, our model uses residual connections to enhance feature extraction capability. The model consists of multiple residual blocks, each of which contains a full connection layer, a Batch Normalization layer, and a Leaky ReLU activation function. The residual connection alleviates the problem of gradient disappearance or gradient explosion that may occur during training of the deep neural network by allowing input from the previous layer to jump to the subsequent layer, thus ensuring the stability and efficiency of the model. The architecture of the model aims to gradually extract higher-level features through layer-by-layer residual blocks, and finally output binary classification results through a fully connected layer, that is, to determine whether the input data point is a malicious node. In this process, the output layer is designed as a single node and the model output is mapped to a probability value between 0 and 1 via the Sigmoid activation function. This design enables the model to effectively deal with potentially malicious nodes labeled from isolated forests while maintaining discrimination against normal nodes.

Figure 3 is a causal chain consisting of many variables. Among them, the causal chain between variables 2, 3, 4, and 6 shows the direct influence relationship between specific variables. Variable 2 has a direct causal effect on variable 3 and variable 4. According to the results in the causal matrix, the influence coefficients of this relationship are 0.1442 (positive) and 0.5622 (positive). This shows that when variable 2 changes, variable 3 will increase correspondingly, variable 4 will increase substantially, and the specific amount of influence is proportional to the amplitude of change in variable 2. The direct causal effect of variable 4 on variable 6 shows a negative relationship with an influence coefficient of −0.1720. This means that when the value of variable 4 increases, the value of variable 6 decreases accordingly, acting as a restraining effect.

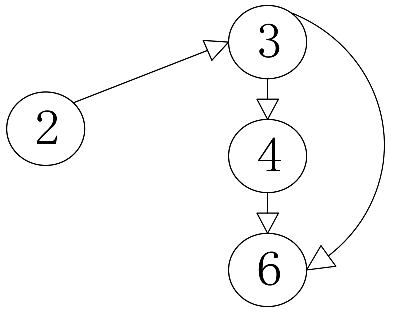

Figure 3. Causal diagram of partial variables.

4. Experiment

This experiment was run on a 13th-generation Intel(R) Core(TM) i7-13700 H 2.40 GHz processor with Windows11 and 32 GB RAM, using the Python 3.8-compiled language and pytorch as a deep learning framework.

The implementation steps include LADIES sampling, which extracts representative subgraphs from large-scale graph data. LiNGAM combines with GAE to fill in the missing data based on causality, as well as the final binary classification prediction of malicious nodes.

4.1. Dataset

In the experimental part of this study, we selected the CADETS [16] dataset from the DARPA TC. The CADETS (Comprehensive, Adaptive, and Detailed Event Tracing System) dataset is part of DARPA's Transparent Computing program, which is designed to capture and analyze events at the operating system level. The dataset contains a wealth of system logs and event records covering a variety of system activities such as file system operations, process creation, and network connections. In this study, we construct a complete data input set by using system logs and event records in the CADETS datasets, interpolating missing data by the LiNGAM graph autoencoder method, and combining threat intelligence data. Finally, we use the residual deep neural network to classify the potential malicious nodes labeled by isolated forest and realize the effective identification and classification of malicious nodes.

4.2. Evolution

To evaluate the filling effect of the model, we calculate the mean square error (MSE) and mean absolute error (MAE) between the filled data and the original data. These metrics are used to quantify the accuracy of filling. The results show that the error between the filled data and the original data is small, indicating that the model can effectively fill in the missing value.

MSE was 0.844 and MAE was 0.667. These results show that the average error of the model in filling in the data is not too large. The mean value of the data after filling is -0.0108 and the standard deviation is 0.938, indicating that the data basically maintains the characteristics of mean value 0 and standard deviation 1 in the standardization process.

The green dots refer to the imputed data, and the following six diagrams (Figure 4) show how the model behaves when different categories of data are inserted. As can be seen from the figure, in most cases, the model captures the trend of the raw data well and reasonably predicts the location of the missing values. The interpolation results show high accuracy in both the wildly fluctuating scenes and those with small changes, indicating that the model has a better understanding of the pattern and structure of the original data.

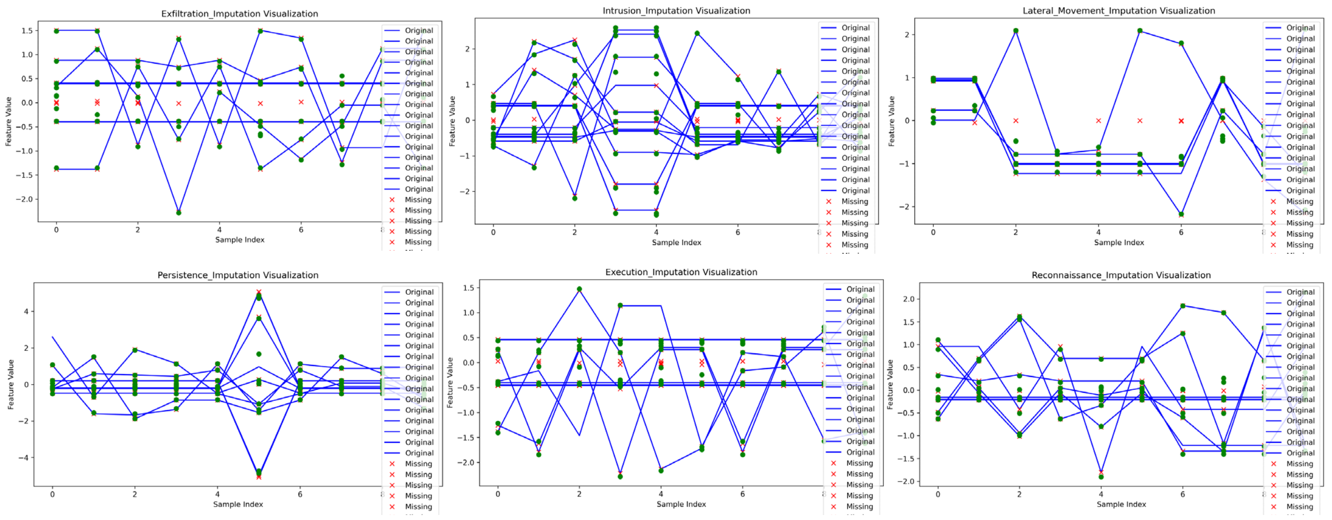

Figure 4. Multi-stage data interpolation graph.

We evaluated the performance of different interpolation methods at different miss rates, mainly measured by MSE and MAE. Because the data missing rate is too high to affect the reliability of the data, we set the data missing rate between 0.04 and 0.2, with a small floating range, and the risk is controllable. As can be seen in Figure 5, the GAIN method is slightly better at lower miss rates but shows slightly less stability than my method as the miss rates increase. In contrast, the traditional mean interpolation and linear interpolation methods have poor performance in terms of error and stability, especially in linear interpolation, the error increases rapidly with the missing rate, and cannot effectively capture the complex trend of data. Although the MICE method shows significant improvement at some deletion rates, its performance is unstable, and overfitting problems may exist especially at low deletion rates. The performance of MLE interpolation is mediocre, and it cannot deal with the nonlinear characteristics of data effectively, which limits its application on complex datasets. However, our method has stable and excellent performance under different miss rates, especially under the condition of high miss rate, its error is still small, demonstrating good learning ability and high robustness to the complex structure of data.

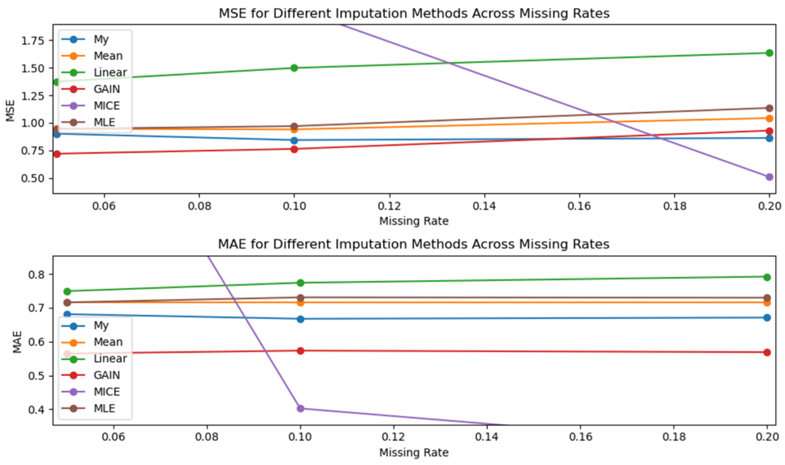

Figure 5. Error value of interpolation method under different missing rates.

The dataset is divided into a training set comprising 80% of the data and a validation set with the remaining 20% to evaluate model performance. We use accuracy, precision, recall, and F1 score as evaluation metrics. Accuracy provides an overall assessment of model performance, precision measures the accuracy of positive predictions, recall assesses the ability to identify positive samples, and the F1 score balances precision and recall. In this study, Transformer [29] is employed for multi-stage APT attack prediction and alert aggregation, chosen for its ability to handle multi-dimensional alert data and capture dependencies across attack stages. The MLP structure is relatively simple, making it well-suited to adapt quickly to changing APT scenarios while maintaining effective detection across stages. RNN captures short-term temporal dependencies in APT attacks, which is especially useful for identifying sequential behaviors in attack patterns. Meanwhile, CNN is advantageous for detecting local features, aiding in uncovering hidden attack behaviors through convolution operations. Figure 6 illustrates the comparative performance of these models based on accuracy, precision, recall, and F1 score. Our model, shown in blue, demonstrates consistently high performance across most training cycles, with a good balance between accuracy and recall, achieving a precision of 0.972 and a recall rate of 0.934, highlighting its reliable identification of malicious samples. However, due to the imbalance between malicious and normal samples, some malicious samples remain undetected. In contrast, the Transformer model also achieves high accuracy and recall, particularly in later training phases, though its performance curve exhibits significant variability, suggesting room for stability improvements. The MLP model excels in accuracy, nearly reaching a perfect score, indicating minimal false positives. However, its recall rate is slightly lower, which impacts its F1 score and may lead to missed positive cases. Both RNN and CNN show comparatively lower performance, with RNN particularly lagging in recall, indicating limitations in capturing all positive cases. CNN performs weakly in both recall and accuracy, which suggests that while it excels in tasks emphasizing local features, it is less suited to this context.

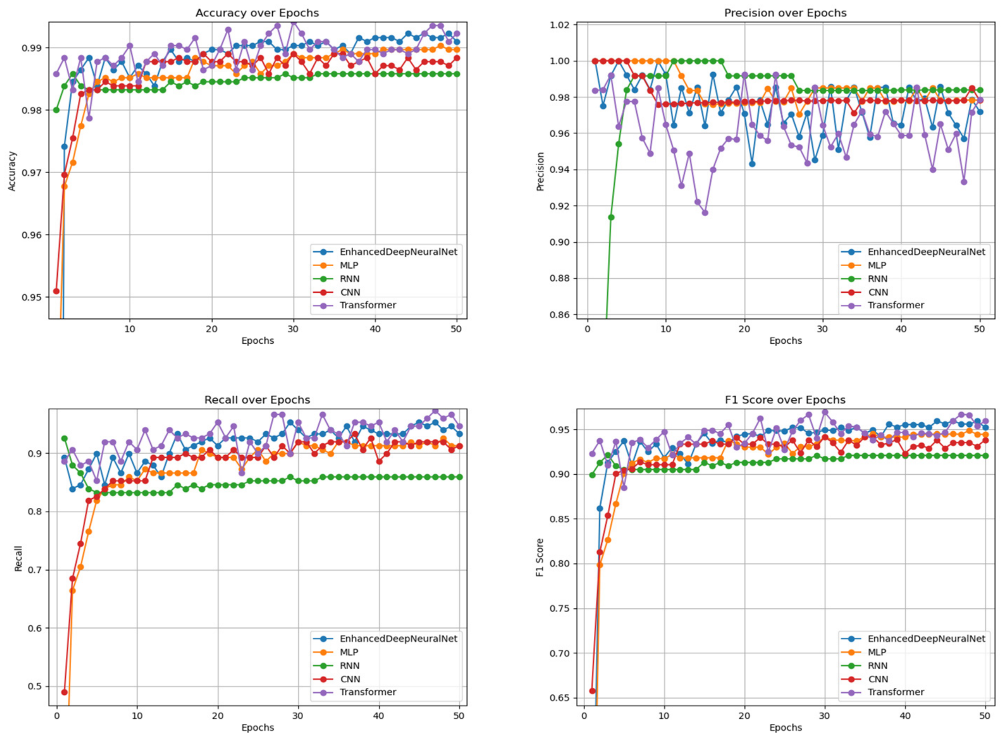

Figure 6. Evaluation.

In general, the model performs well in the detection of malicious nodes, especially in the accuracy and recall rate, which shows its adaptability to the multi-stage characteristics of APT attacks and complex causal association analysis ability. However, differences in the performance of different models also reveal directions for improvement, especially in model stability and adaptability to different attack modes.

5. Conclusions

In this paper, we propose a comprehensive deep learning framework for dealing with complex graph-structured data and missing data by combining causal discovery and graph autoencoder models. Additionally, the framework integrates the Isolation Forest algorithm for potential malicious node detection, followed by classification and validation using a deep neural network. Experimental results demonstrate that, even within complex and dynamic data environments, our interpolation method performs effectively in handling missing data, maintaining stable performance. The accuracy of malicious nodes detection remains high across various experiments, which proves the effectiveness of the model in real scenarios. This method is not only effective in predicting APT attacks, but is also applicable to other situations with missing data, whether it is in the field of social network analysis or other network security, it can effectively fill in data and identify anomalies while maintaining causality. To further improve model performance, future research can focus on enhancing the robustness and adaptability of the model through effective data augmentation and the integration of domain knowledge. Advanced data augmentation techniques would increase training data diversity, improving the model's ability to recognize a wide range of attack patterns. Incorporating domain knowledge will also enable the model to better identify complex attack patterns by capturing key characteristics and potential anomalies within complex, dynamic data environments. Together, these improvements will support efficient detection capabilities in an evolving threat landscape and advance the development of security detection technologies.

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

Funding: This study was supported by the Natural Science Foundation of Jiangsu Province (No. BK20220562), the Open Fund for the Key Laboratory of Flying Internet at Civil Aviation University of China (No. MHFLW202304), and the Xinjiang Uygur Autonomous Region Natural Science Foundation project (No. 2024D01A40).

Data Availability Statement: Data are contained within the article.

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

References

1. Mittal, M.; Kumar, K.; Behal, S. Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Comput.* **2023**, *27*, 13039–13075. [CrossRef] [PubMed]
2. Quintero-Bonilla, S.; Martín del Rey, A. A new proposal on the advanced persistent threat: A survey. *Appl. Sci.* **2020**, *10*, 3874. [CrossRef]
3. Yang, H.; Wang, Z.; Zhang, L.; Cheng, X.C. A Multi-Protocol Botnet Detection Method for IoT. *Acta Electron. Sin.* **2023**, *51*, 1198–1206. [CrossRef]
4. Karantzas, G.; Patsakis, C. An empirical assessment of endpoint detection and response systems against advanced persistent threats attack vectors. *J. Cybersecur. Priv.* **2021**, *1*, 387–421. [CrossRef]
5. Xie, L.; Li, X.; Yang, H.; Zhang, L. A Multi-stage APT Attack Detection Method Based on Sample Enhancement. In *International Symposium on Cyberspace Safety and Security*; Springer International Publishing: Cham, Switzerland, 2022; pp. 209–216.
6. Stojanović, B.; Hofer-Schmitz, K.; Kleb, U. APT datasets and attack modeling for automated detection methods: A review. *Comput. Secur.* **2020**, *92*, 101734. [CrossRef]
7. Neuschmied, H.; Winter, M.; Stojanović, B.; Hofer-Schmitz, K.; Božić, J.; Kleb, U. Apt-attack detection based on multi-stage autoencoders. *Appl. Sci.* **2022**, *12*, 6816. [CrossRef]
8. Wilkens, F.; Ortmann, F.; Haas, S.; Vallentin, M.; Fischer, M. Multi-stage attack detection via kill chain state machines. In *Proceedings of the 3rd Workshop on Cyber-Security Arms Race*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 13–24.
9. Zhou, P.; Zhou, G.; Wu, D.; Fei, M. Detecting multi-stage attacks using sequence-to-sequence model. *Comput. Secur.* **2021**, *105*, 102203. [CrossRef]
10. Takey, Y.S.; Tatikayala, S.G.; Samavedam, S.S.; Eswari, P.L.; Patil, M.U. Real time early multi stage attack detection. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 1, pp. 283–290.
11. Annadani, Y.; Pawlowski, N.; Jennings, J.; Bauer, S.; Zhang, C.; Gong, W. BayesDAG: Gradient-Based Posterior Inference for Causal Discovery. In Proceedings of the 37th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Advances in Neural Information Processing Systems. Volume 36.
12. Fan, S.; Wang, X.; Shi, C.; Cui, P.; Wang, B. Generalizing graph neural networks on out-of-distribution graphs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2023.
13. Kaltenpoth, D.; Vreeken, J. Nonlinear causal discovery with latent confounders. In Proceedings of the PMLR: 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; pp. 15639–15654.
14. Sun, B.; Sun, J.; Pham, L.H.; Shi, J. Causality-based neural network repair. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 25–27 May 2022; pp. 338–349.
15. Yang, J.; Zhang, Q.; Jiang, X.; Chen, S.; Yang, F. Poirot: Causal correlation aided semantic analysis for advanced persistent threat detection. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3546–3563. [CrossRef]
16. Strnad, A.; Messiter, Q.; Watson, R.; Carata, L.; Anderson, J.; Kidney, B. *Casual, Adaptive, Distributed, and Efficient Tracing System (Cadets)*; Technical Report; BAE Systems Burlington United States: Burlington, MA, USA, 2019.
17. Akbar, K.A.; Wang, Y.; Ayoade, G.; Gao, Y.; Singhal, A.; Khan, L.; Thuraisingham, B.; Jee, K. Advanced Persistent Threat Detection Using Data Provenance and Metric Learning. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 3957–3969. [CrossRef]
18. Zhu, T.; Yu, J.; Xiong, C.; Cheng, W.; Yuan, Q.; Ying, J.; Chen, T.; Zhang, J.; Lv, M.; Chen, Y.; et al. Aptshield: A stable, efficient and real-time apt detection system for linux hosts. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 5247–5264. [CrossRef]
19. Huang, G. Missing data filling method based on linear interpolation and lightgbm. *J. Phys. Conf. Ser.* **2021**, *1754*, 012187. [CrossRef]
20. Yu, L.; Liu, L.; Peace, K.E. Regression multiple imputation for missing data analysis. *Stat. Methods Med. Res.* **2020**, *29*, 2647–2664. [CrossRef] [PubMed]
21. Kim, J.K.; Shao, J. *Statistical Methods for Handling Incomplete Data*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2021.

22. Yoon, J.; Jordon, J.; Schaar, M. Gain: Missing data imputation using generative adversarial nets. In Proceedings of the PMLR: International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5689–5698.
23. Peng, Z.; Huang, W.; Gu, S.; Xie, L.; Wang, Y.; Jiao, J.; Ye, Q. Conformer: Local features coupling global representations for visual recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 367–376.
24. Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Sci. Rep.* **2018**, *8*, 6085. [CrossRef] [PubMed]
25. Zhu, H.; Lin, Y.; Liu, Z.; Fu, J.; Chua, T.S.; Sun, M. Graph neural networks with generated parameters for relation extraction. *arXiv* **2019**, arXiv:1902.00756.
26. Zou, D.; Hu, Z.; Wang, Y.; Jiang, S.; Sun, Y.; Gu, Q. Layer-dependent importance sampling for training deep and large graph convolutional networks. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Advances in Neural Information Processing Systems. Volume 32.
27. Che Mat, N.I.; Jamil, N.; Yusoff, Y.; Mat Kiah, M.L. A systematic literature review on advanced persistent threat behaviors and its detection strategy. *J. Cybersecur.* **2024**, *10*, tyad023. [CrossRef]
28. Ikeuchi, T.; Ide, M.; Zeng, Y.; Maeda, T.N.; Shimizu, S. Python package for causal discovery based on LiNGAM. *J. Mach. Learn. Res.* **2023**, *24*, 1–8.
29. Wang, W.; Yi, P.; Jiang, J.; Zhang, P.; Chen, X. Transformer-based framework for alert aggregation and attack prediction in a multi-stage attack. *Comput. Secur.* **2024**, *136*, 103533. [CrossRef]