Hindawi
Security and Communication Networks
Volume 2021, Article ID 8986243, 18 pages
https://doi.org/10.1155/2021/8986243

*Research Article*

# Improving the Accuracy of Network Intrusion Detection with Causal Machine Learning

**Zengri Zeng,[1,2] Wei Peng ,[1] and Baokang Zhao[1]**

[1]*College of Computer, National University of Defense Technology, Changsha 410073, China*
[2]*Information Institute, Hunan University of Humanities, Science and Technology, Loudi 417000, China*

Correspondence should be addressed to Wei Peng; wpeng@nudt.edu.cn

Received 28 July 2021; Revised 9 September 2021; Accepted 4 October 2021; Published 3 November 2021

Academic Editor: Zhiyuan Tan

In recent years, machine learning (ML) algorithms have been approved effective in the intrusion detection. However, as the ML algorithms are mainly applied to evaluate the anomaly of the network, the detection accuracy for cyberattacks with multiple types cannot be fully guaranteed. The existing algorithms for network intrusion detection based on ML or feature selection are on the basis of spurious correlation between features and cyberattacks, causing several wrong classifications. In order to tackle the abovementioned problems, this research aimed to establish a novel network intrusion detection system (NIDS) based on causal ML. The proposed system started with the identification of noisy features by causal intervention, while only the features that had a causality with cyberattacks were preserved. Then, the ML algorithm was used to make a preliminary classification to select the most relevant types of cyberattacks. As a result, the unique labeled cyberattack could be detected by the counterfactual detection algorithm. In addition to a relatively stable accuracy, the complexity of cyberattack detection could also be effectively reduced, with a maximum reduction to 94% on the size of training features. Moreover, in case of the availability of several types of cyberattacks, the detection accuracy was significantly improved compared with the previous ML algorithms.

## 1. Introduction

Cyberattacks [1] refer to offensive actions to alter, disrupt, deceive, degrade, or destroy computer systems, networks, information, or programs in these systems. In recent years, the high frequency of cyberattacks has posed severe threats to the network security and even the national security, leading to a significant decline in network performance and service interruption. Hence, a great number of protection mechanisms [2, 3] have been proposed and deployed, such as firewalls, antiviruses, and malware detection software. However, these countermeasures have been proved insufficient to provide a complete protection against the cyberattacks in the modern network environments.

Although firewalls can provide rule-based network protection, more intelligent mechanisms are required to detect advanced network intrusion in high volume of traffic data. To this end, several network intrusion detection systems (NIDSs) [4–6] have been designed using ML methods. A NIDS can provide real-time data on network traffic and send out an instant alarm or block suspicious activities if a network attack is detected. ML methods are widely utilized in NIDSs to detect a network's anomalies mainly through extracting features of traffic data.

Although ML-based NIDSs have shown to be robust in real-time traffic monitoring, their accuracy and efficacy are still compromised by the imprecise features, which are greatly dependent on a human's experience. Meanwhile, a fixed feature set may not be appropriate for detecting different types of network intrusions, as some features may be redundant or unrelated, which may slow down the ML process. Therefore, it is essential to explore the best features [7] to increase the accuracy of a detection system.

To overcome the abovementioned barriers, application of causal ML methods in NIDSs is proposed in this paper. Traffic features can be classified into two classes: causal features and noisy features. Causal features are those features, which have causal relationships with a network intrusion. That is, these features are caused by cyberattacks. When cyberattacks are launched, these features become abnormal. While the cyberattacks are stopped, these features become normal. Traditional distributed denial-of-service (DDoS) attacks exhaust the bandwidth, central processing unit (CPU) power, or memory of the victim host by flooding an overwhelming number of packets from thousands of compromised computers (zombies) to deny legitimate flows. The most frequent DDoS attacks mainly consist of flooding with a huge volume of traffic data and consuming network resources, such as bandwidth, buffer space at the routers, CPU power, and recovery cycles of the target server. Noisy features have no causal relationship with a cyberattack, although they may have a statistical-based correlation [8]. Noisy features can degrade detection performance because they may disrupt a detection system in real deployment.

To distinguish noisy features from causal features in NIDSs, we present two causal ML methods for NIDSs, including causal intervention and counterfactual reasoning.

The main contributions of this paper include

(i) We propose a novel causal ML-based NIDS. With establishing a causal link between cyberattacks and traffic features through causal intervention, noisy features can be identified and removed.

(ii) A counterfactual detection algorithm based on the Bayesian Network (BN) is developed to classify cyberattacks based on causal features.

(iii) The performance of the causal ML-based NIDS is evaluated using CICIDS19, UNSW-NB15, and NSL-KDD datasets. The experiment results confirmed the effectiveness of the proposed approach.

This paper is organized as follows.

Section 2 provides a brief discussion on the existing relevant studies on NIDSs and their limitations, as well as a summary on the contributions of this study. Section 3 presents a detailed discussion on the theories and governing equations of the different deployment techniques. Section 4 presents a novel causal ML-based NIDS. Section 5 mainly discusses on the experimental results. And, Section 6 summarizes the main achievements of this research.

## 2. Literature Review

As one of the important areas in computer science and network security, intrusion detection based on ML [9–11] is a hotspot. Numerous scholars [12–15] have already carried out a variety of explorations on this topic. Tang et al. [16] established a deep neural network model of NIDSs, and the model was trained by the NSL-KDD dataset. Their model showed a robustness for detecting flow-based anomalies in software-defined networking (SDN). Daya et al. [17] proposed BotChase, a two-phased graph-based bot detection system, leveraging both unsupervised and supervised ML. The first phase pruned presumable benign hosts, while the second phase achieved bot detection with high precision. The literature [18] on NSL-KDD dataset aimed to propose an adaptive ensemble learning model to develop a multitree algorithm with an accuracy of 84.2%.

As reported previously, optimization of the size of training features is worthy of investigation. Importantly, irrelevant features in a dataset could undermine accuracy of a model and increase training time required for the establishment of a model. Thus, to determine the optimum training size, numerous explorations have been conducted. Feature selection [11, 19–22], a process of selecting the most relevant features by manual or algorithms, has been used to reduce the time and space complexity of model construction. Hadeel et al. [23] proposed a wrapper feature selection algorithm for intrusion detection. This method uses a dove-inspired optimizer to implement the feature selection, and the binarizing algorithm of the proposed cosine similarity method showed a faster convergence speed and a higher accuracy than the sigmoid method. Another research [24] developed a feature selection model, which combined ID3 classifier algorithm and BEES algorithm. In this model, the BEES algorithm was used to generate the desired feature subset. Chung and Wahid [25] introduced a new simplified version of particle swarm optimization for feature selection, constituting a local search strategy to speed up the feature selection process by finding the optimal neighborhood solution. The algorithm could reduce the features used to represent network traffic behavior in KDDCUP99 dataset from 41 to only 6, and the accuracy reached 93.3%. However, the method mentioned above could only select features based on relevance, and some noisy features may affect the detection accuracy.

In addition to the size of training features, correct classification of cyberattacks is also of great importance in the existing studies. The existing algorithms for NIDSs based on ML or feature selection are all on the basis of correlation between features and cyberattacks to realize the classification. This correlation causes several wrong classifications due to the existence of a large number of spurious correlations [26]. In order to solve this problem, causal reasoning [27–32] is frequently utilized to solve the spurious correlations. At present, causal reasoning mainly adopts two models [33]: sStructural causal model (SCM) [34] and potential outcome model (POM) [35]. A SCM is made of endogenous (manifest) and exogenous (latent) variables. The POM provides the causal effects [36] through mathematical definitions. However, conducting randomized trials [37] with both SCM and POM is expensive, time-consuming, and sometimes unethical. Additionally, its accuracy is low, owing to insufficient consideration about the influences of exogenous variables (a variable outside the cyberattack model, which affects the cyberattack model but is not affected by the cyberattack model) [26] and noisy factors on the causal features.

Based on the deficiencies of the abovementioned algorithms, this paper starts from the decoupling of the correlation of features and the classification of types of cyberattacks under counterfactual scenarios to achieve a high accuracy in the detection of cyberattacks. The counterfactual model is based on the BN, which can model relationships among hundreds of cyberattacks and features. Firstly, the correlation of features is decoupled through causal intervention, and noisy features that do not affect the detection outcome are deleted. Secondly, based on the retained causal features, the most relevant types of labels are selected, and then, the counterfactual detection algorithm is implemented to find out the unique label. For instance, given evidence $\varepsilon = e$ and some hypothetical interventions, the likelihood that we observed a different outcome $\varepsilon = e'$ through the counterfactual detection algorithm is calculated. Then, the expected number of anomalous features is calculated to identify the highest likelihood of cyberattacks in the counterfactual scenario [26].

## 3. Preliminaries

In this section, we present a brief introduction about causal reasoning.

*3.1. Strong Spurious Correlations.* Traditional ML is driven by the association, and it is difficult to achieve consistent prediction for unknown test datasets. Traditional ML will find noncausal (noise) features in association mining, such as the relationship between risk factors and abnormal features, and such strong spurious correlations will be used for the prediction.

For example, risk factor $R$ will cause DDoS attacks in Figure 1, for instance, $X_1$, $X_2$, and $X_3$, and $X_1$ and $X_2$ will cause abnormalities of traffic feature $Y_1$ and $Y_2$. If $X_1$ and $X_2$ have not been observed or counted in the prior data, risk factor $R$ will inevitably lead to the appearance of $X_3$, $Y_1$, and $Y_2$. If the calculation is based on the correlation algorithm only, the conclusion that $X_3$ is the cause of $Y_1$ and $Y_2$ may be completely wrong.

A classic New England Medicine paper on chocolate and the Nobel Prize [38] explains such strong spurious correlations. According to the paper, the more chocolate a country consumes, the more Nobel Prizes it will win. This conclusion is very absurd at the first glance, but what is wrong with the conclusion based on relevant facts? Statistical analysis of the data shows that there is indeed a linear relationship between a country's chocolate sales and the number of Nobel Prizes it has won. However, the causal analysis indicates that there is only a strong spurious correlation between chocolate sales and the number of Nobel Prizes.

*3.2. Definitions.* It is supposed that $Y = \{C, V\}$ is the traffic feature set, where $C$ is a causal feature set and $V$ indicates a noisy feature set ($V = Y\backslash C$). $X \in \{0, N\}$ represents a network attack.

As noisy features have no causal relationships with network intrusions, the conditional probability $P\ (X|Y)$ satisfies the following condition [8]:

$$P(X|Y) = P(X|C, V) = P(X|C). \quad (1)$$

Although there is no causality between $X$ and $V$, they may show a strong correlation in the statistical data (Figure 2(b)). If the spurious relationship is not distinguished from causation, it may lead to errors in real-world data distributions, even if the ML model is trained well.

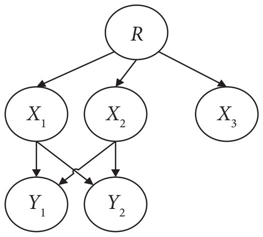


Figure 1: Spurious correlation features.

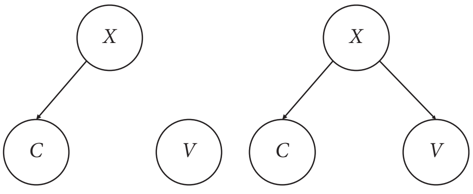
To define causality, if other conditions do not change, changing $X$ can cause a change in $Y$; thus, there is a causality between $X$ and $Y$. If $X$ and $Y$ can be measured, then the causal relationship of $X$ and $Y$ can be calculated by changing the values of $X$ and $Y$. If the magnitude of the causal relationship between $X_1$ and $Y$ is stronger than that between $X_2$ and $Y$, it is considered that $X_1$ causes $Y$.

In general, cyberattacks cause the anomaly of data traffic features, as shown in Figure 3. For the sake of a simpler analysis, exogenous variables are ignored. As mentioned earlier, if other conditions remain unchanged, the change of $\{Y_1, Y_2, \ldots, Y_n\}$ may lead to the change of $X$, which indicates that there is a causal relationship between $\{Y_1, Y_2, \ldots, Y_n\}$ and $X$. Meanwhile, it is equivalent to the fact that $X$ is the cause, and $\{Y_1, Y_2, \ldots, Y_n\}$ is the effect.

*3.3. SCM.* The detection models which will be used in our experiments are BN models which show the relationships between cyberattack, risk factors, and traffic features. BNs are an increasingly popular modelling technique in cybersecurity [39], especially due to their capability to overcome data constraints (it is impossible to learn causality between variables). In BNs, the probability is interpreted as a degree of confidence. As shown in Figure 4, in the 3-layer BN model, the traffic features are influenced by corresponding cyberattacks, where $Z$ is the risk factor of the network being attacked, $X$ denotes the type of cyberattack, and $Y$ represents the traffic features. In the noisy-OR model, $Y = (X_1 \vee X_2 \vee, \ldots, \vee X_n)$, and as long as there is an attack type $X_i = 1$, then $Y = 1$. This pattern (Figure 4) can be extended to a further complex network model with more layers.

In the causal inference, BN is replaced by a more basic SCM. Existing BNs can be expressed as a SCM [40, 41]. This SCM consists of three components [42]: a graphical model, a structural equation, and a counterfactual and intervention logic.

The key characteristic of SCMs is that they represent each variable as deterministic functions of their direct causes together with an unobserved exogenous "noise" term, which itself represents all causes outside of our model. For example, in a network without cyberattacks, some traffic features may be abnormal, which is due to unobserved exogenous variables. If an unobserved exogenous variable $u = \{u_1, u_2, \ldots, u_n\}$ is specified, the causal Markov blanket (for complete random variable UR and a given set of variables $X \in \mathrm{UR}$ and $\mathrm{MB} \subset \mathrm{UR}(X \notin \mathrm{MB})$, if $X \perp \{\mathrm{UR} - \mathrm{MB} - \{X\}\}|\mathrm{MB}$, the minimum variable set MB that can meet the above conditions is a Markov blanket with $X$) condition [26, 42, 43] will be satisfied.



(a) (b)

Figure 2: The relationship between cyberattacks and extracted features. (a) $X \perp V$. (b) $X \longrightarrow V$.

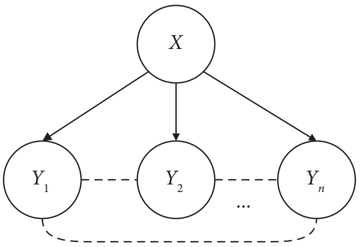
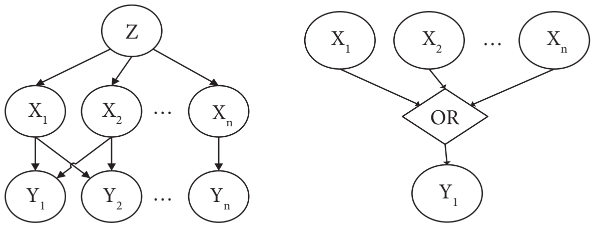


Figure 3: Causality between cyberattacks and extracted features.



A 3-layer BN model A noisy-OR model

Figure 4: Schematic illustration of SCMs.

*Assumption 1.* It is assumed that the observed variable is $Y = \{Y_1, Y_2, \ldots, Y_n\}$ in the SCM of the directed acyclic graph [42]; its parent variables can be regarded as $u$ v pa $(Y)$; thus, $Y = f$ {pa $(Y)$, $u$} can be achieved. For each variable $Y$, the parent variable $X$ (i.e., $X = \text{pa}\ (Y)$) in the model has a noise term $u_y$ with an unknown distribution $P(u_y)$, such that

$$P(Y = y|X = x) = \sum_{u_y:\, f\left(x,u_y\right)} P\left(U_y = u_y\right). \tag{2}$$

*Assumption 2.* In the noisy-OR model [39], it is assumed that the probability that any variables $Y$ may behave as normal $(Y = 0)$ due to noisy variables in a network attack $(X_i = 1)$ is $L_{X_i,Y}$. It is assumed that the variables $Y$ are independent of each other, and then,

$$L_{X_i,Y} = \prod_{i=1}^{n} P\left(Y = 0|X_i = 1, \wedge_{j,i\neq j} X_j = 0\right). \tag{3}$$

For instance, the network devices are installed with antivirus software or firewalls; thus, some traffic features may not produce abnormalities.

*3.4. Causal Intervention.* The causal detection problem (magnitude of the causality, feature selection, unobserved exogenous variables, and noisy variables) can be addressed by a causal intervention that is called "do-operation."

*Definition 1* (do-operation). The postintervention distribution resulting from the action $(Y = y)$ is given by equation (4) [40]:

$$P(X = x|\text{do}(Y = y)) = P_m(x|y). \tag{4}$$

The do-operator of causal intervention signifies that we are dealing with an intervention, rather than a passive observation. The subscript $m$ is used to represent the modified probability distribution. From the perspective of probability distribution, $P\ (X = x|Y = y)$ represents the probability of $X = x$ corresponding to the part of $Y$ among all the values that $Y = y$, and $P\ (X = x|\text{do}\ (Y = y))$ represents the probability that all $Y$ are fixed to $y$ and then $X = x$. Intervention changes the distribution of the original data, while conditional variables do not change the distribution of the original data.

*3.5. Counterfactual Detection [26].* Counterfactuals enable us to quantify how well a cyberattack (i.e., $X = 1$) explains anomalous features by determining the likelihood that the features may not be presented during intervention, thereby switching to the cyberattacks by setting do $(X = 0)$, as given by the counterfactual probability $P\ (Y = 0|Y = 1, \text{do}\ (X = 0))$. If the probability is high, $X = 1$ is a good causal explanation of the anomalous features. It should be noted that this probability refers to two contradictory states of $Y$, and thus, it cannot be represented as a standard posterior probability.

The principles for counterfactual detection of cyberattacks are as follows [26, 37]:

(1) The likelihood that a cyberattack causes an anomalous feature should be proportional to the posterior likelihood of that attack

(2) A cyberattack $X$, which cannot cause an anomalous feature, cannot constitute a causality between features and attacks

(3) A type of cyberattack, which causes a greater number of anomalous features, should be more likely to have a causality to these features

## 4. A Novel Causal ML-Based NIDS

In this section, the causal ML-based NIDS (CMLN) framework and time complexity will be introduced.

*4.1. Framework.* This study aims to develop a novel causal ML-based NIDS. As illustrated in Figure 5, the proposed framework is divided into four main stages. The first stage is data preprocessing, consisting of $Z$-score, Min-Max, and deletion of the incorrect and fuzzy row datasets. The purpose of this step is to improve the performance of the training model and reduce the class imbalance problem [26] that often appears in network traffic data. Hence, data should be initially encoded with $Z$-score to transform any categorical features into numerical ones. Then, the value of a normal feature is equal to 0 and that of an anomalous feature is a positive integer [37, 40] in causal reasoning; thus, it needs to be normalized to a natural number. At the end, incorrect and fuzzy row datasets should be removed to reduce the size of training dataset and improve the accuracy of validation dataset.

The second stage of the framework is the processing of selected features, which reduces the number of features required for ML models and counterfactual detection algorithm. Firstly, although the noisy features may have a correlation with the causal features, they have no causal effect on the classified outcomes. The causal relationship between the features and cyberattacks can be identified through causal intervention. Then, the noisy features are deleted, and only few features can be retained. This not only reduces the time required for the model classification but also reduces the time required for training without sacrificing other functions.

Two correlated variables have a causal relationship, while two uncorrelated variables have no causal relationship. ML algorithms are involved in the third stage of the framework to select several classes of labels. The labels with the largest correlation are selected as the reference labels of the fourth stage, which can also reduce the complexity of counterfactual detection algorithm. Therefore, it is necessary for the counterfactual detection algorithm to calculate the expected anomalous features of $K$ cyberattacks, without calculation of the expected anomalous features of $M$ cyberattacks ($K$ includes reference labels selected by the ML algorithm, and $M$ covers all labeled cyberattacks).

In the fourth stage, according to the causality, it can be determined whether the results of the counterfactual detection algorithm will change or not when certain preconditions change and then provide the basis for the counterfactual judgment according to the magnitude of the causality effect. Given the evidence $\varepsilon = e$ and an intervention all cyberattacks are switched except for $X_a$ in counterfactual. Next, the number of expected anomalous features $E\,(X_k, \varepsilon)$ is calculated ($X_a$ belongs to $X_k$ and $X_k$ includes reference labels selected by the ML algorithm). Finally, with obtaining the largest value of $E\,(X_k, \varepsilon)$, the most likelihood of a cyberattack is $X_k$.

With the joint action of these four stages, the causal ML-based NIDS could ensure a high accuracy in the detection of anomalous features when the types of cyberattacks are increased.

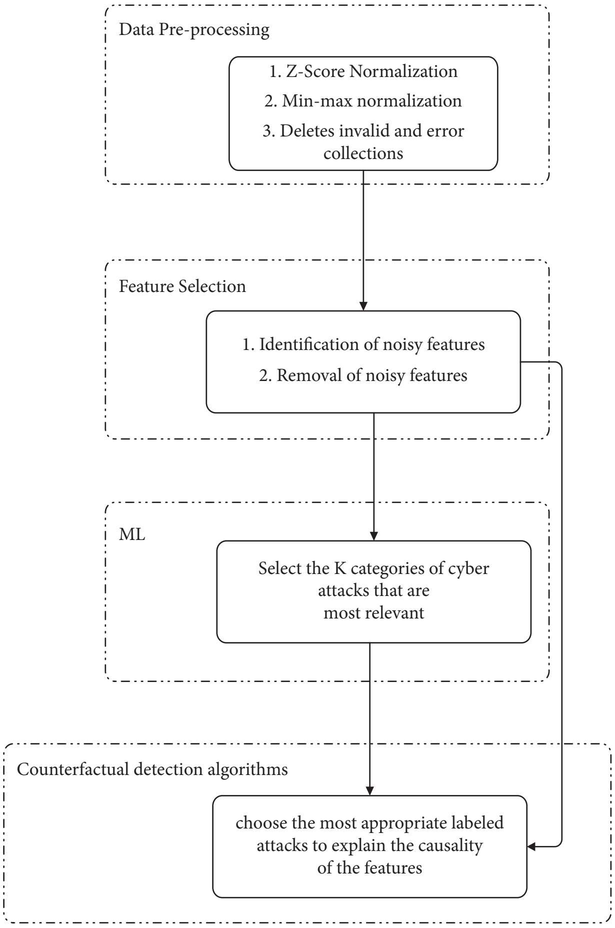

Data Pre-processing
1. Z-Score Normalization
2. Min-max normalization
3. Deletes invalid and error collections
Feature Selection
1. Identification of noisy features
2. Removal of noisy features
ML
Select the K categories of cyber attacks that are most relevant
Counterfactual detection algorithms
choose the most appropriate labeled attacks to explain the causality of the features

Figure 5: The framework of the proposed causal ML-based NIDS.

*4.2. Data Preprocessing.* Performing data normalization by using the $Z$-score, positive integerization by using the Min-Max normalization, and deletion of the incorrect and fuzzy row datasets are covered in the data preprocessing stage.

*4.2.1. Z-Score Normalization.* $Z$-score normalization [44, 45] of the data is initially carried out. The most common standardization method is $Z$-score standardization, which is also known as standard deviation standardization. The main purpose of $Z$-score is to transform features of different magnitudes into the same magnitude and measure the features with the calculated $Z$-Score value to ensure comparability of them. This method presents the mean and standard deviation of the original data to conduct data standardization. The processed data conform to the standard normal distribution, that is, the mean value is 0, the standard deviation is 1, and the transformation function is

$$Y_{z\text{score}} = \frac{Y_{\text{inst}} - U}{\delta}, \tag{5}$$

where $Y_{\text{inst}}$ is the initialized feature value, $U$ denotes the mean feature vector, and $\delta$ is the standard deviation.

*4.2.2. Min-Max Normalization.* Min-Max normalization [46], also known as deviation normalization, is a linear transformation of the original data, with max being the maximum and min the minimum of the sample data. In the counterfactual detection algorithm, the value of a normal feature is 0 and that of an anomalous feature is a positive integer; thus, it needs to be normalized to a natural number. Data normalization is a necessary step, in which each value needs to be extended to an appropriate range. This process helps eliminate large deviations in features:

$$\psi_{ij} = \text{Round}\left[\left(\frac{Y_{ij} - \min\left(Y_j\right)}{\max\left(Y_j\right) - \min\left(Y_j\right)}\right) * N\right], \qquad (6)$$

where $\psi_{ij}$ indicates the normalized value of $Y_{ij}$ with the range of *0* to $N$ in the integer form, min ($Y_j$) represents the minimum value of the $j$th feature, and max ($Y_j$) is the maximum value of the $j$th feature.

*4.2.3. Removal of Incorrect and Fuzzy Row Sets.* There are features with empty values in the row of features or the label corresponding to this row of features without a dependency on a normal attack category in the intrusion detection dataset. Thus, this row is an invalid or incorrect row set. Alternatively, the row of features is corresponding to multiple types of cyberattacks (such as features [0, 1, 1, 1] corresponding to two types of cyberattacks, DDos, and exploits); as a result, this row is a fuzzy row set [47]. The incorrect and fuzzy sets cannot be labeled by ML algorithms. Therefore, the incorrect and fuzzy sets need to be deleted in the data preprocessing stage, and only a certain subset is left, in which the row features and label have one-to-one definite correspondences (e.g., the row of features [0, 1, 1, 1] is uniquely corresponding to a DDoS), so as to improve the robustness of the causal ML-based NIDS.

*4.3. Feature Selection.* If some features are irrelevant to the cyberattacks and they have no causal effect on the classified outcomes [26], these features are therefore noisy features. Normally, manually matching of features can be used directly to eliminate the impacts created by noisy features on the classified outcomes. However, when it comes to training by ML algorithms, a classifier will constantly fit these features, leading to a spurious correlation between noisy features and cyberattacks. Ultimately, the performance of the classifier could be impaired. This mainly involves the effects of causality on each feature, and calculation is carried out to assess the effects of causality. Consequently, the noisy features are distinguished and deleted based on the effects of causality. Hence, the best combination of causality-based features could be made.

*4.3.1. Identification of Noisy Features.* As shown in Figure 6, there are various relationships between cyberattack $X$ and feature $Y$ under the general fact. If the causal relationship and direction between these two parameters are not clarified, the judgment of the type of cyberattack may be influenced. As displayed in Figure 6(b), it is assumed that $Y_i$ and $Y_j$ have a mutually causal relationship, and the anomaly of one feature will lead to the anomaly of the other. Therefore, there may be a wrong conclusion if the anomalous feature $Y_j$ is considered to be caused by the cyberattack $X$.

According to this hypothesis, reversal of the causal direction of the fact between cyberattack $X$ and feature $Y$ is illustrated in Figure 6(c). Therefore, feature $Y$ can be interfered, and the causal relationship between $Y$ and $X$ can be worked out according to changes of the expected value of $X$, which is formulated as [48]

$$\begin{aligned} E[X|\text{do}(Y_i)] &= \sum_x xP(x|\text{do}(Y_i = y_i)) \\ &= \sum_x \sum_{y_j} xP\left(x|y_i, y_j\right)P\left(y_j|y_i\right). \end{aligned} \qquad (7)$$

If the conditions between $Y$ and $X$ satisfy the following rules, respectively, equation (7) can be written as (8)–(15) [43].

*Rule 1.* If $Y_i$ and $Y_j$ are independent, then

$$\begin{aligned} E[X|\text{do}(Y_i)] &= \sum_x xP(x|\text{do}(Y_i = y_i)) \\ &= \sum_x \sum_{y_j} xP\left(x|y_i, y_j\right)P\left(y_j\right). \end{aligned} \qquad (8)$$

*Proof.* In the statistical model, the calculation formula of the joint distribution is

$$P(x_1, x_2, x_3, \ldots, x_n) = P(x_1) \prod_{i=2}^{n} P(x_{i-1}, \ldots, x_1). \qquad (9)$$

According to the Markov blanket [26, 43], in a directed acyclic graph, given the parent node of $X$, $X$ is independent of nonchild nodes of its parent. Hence, the abovementioned formula can be abbreviated as

$$P(X) = \prod_{i\varepsilon n} P(x_i|\text{Pa}(x_i)), \qquad (10)$$

where Pa ($x_i$) represents the parent node of $x_{\text{i}}$. This formula also represents a BN. As depicted in Figure 6(c), it can be simplified as follows:

$$P\left(x, y_i, y_j\right) = P\left(x|y_i, y_j\right)P\left(y_i|y_j\right)P\left(y_j|y_i\right). \qquad (11)$$

According to the truncated factorization,

$$P\left(x, y_i|\text{do}\left(y_j\right)\right) = P\left(y_j\right)P\left(x|\text{do}\left(y_i\right), y_j\right). \qquad (12)$$

Marginalized $y_j$:

$$P\left(x|\text{do}\left(y_j\right)\right) = \sum_{y_j} P\left(x|y_i, y_j\right)P\left(y_j\right). \qquad (13)$$

Thus, $E[X|\text{do}(Y_i)] = \sum_x \sum_{y_j} xP(x|y_i, y_j)P(y_j)$ □

*Rule 2.* If $Y_i$ and $X$ are independent, then

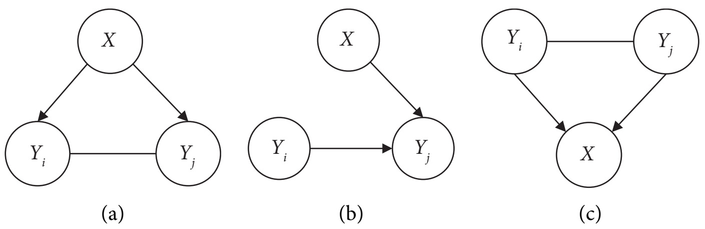


Figure 6: The simplified illustration of the influences of features on cyberattacks.

$$E[X|\mathrm{do}(Y_i)] = \sum_x xP(x) = \sum_x \sum_{y_j} xP(x|y_j)P(y_j|y_i). \tag{14}$$

*Rule 3.* If $Y_i$ is independent of $Y_j$ and $X$, thus,

$$E[X|\mathrm{do}(Y_i)] = \sum_x xP(x) = \sum_x \sum_{y_j} xP(x|y_j)P(y_j). \tag{15}$$

The causal effect [49] can be calculated by measure $E$ of $X$ and $Y$:

$$E = E[X|\mathrm{do}(Y=1)] - E[X|\mathrm{do}(Y=0)]. \tag{16}$$

*Definition 2* (noisy features). As for noncausal features, if E/N (N is the size of training dataset) is less than the threshold $\delta$ ($\delta \leq 0.01$), there will be no causal relationship [50] between $X$ and $Y$. Thus, these features can be considered as noisy features, and they should be deleted in the dataset.

*4.3.2. Removal of Noisy Features.* The causal interventions were performed for all features, as shown in Figure 7. In the process of feature selection, only those features that have a causal relationship with the labeled attacks will be selected. As illustrated in Figure 7, the correlation between features is hidden.

If there is no causal relationship between $\{Y_1, Y_3, \ldots, Y_{n-1}\}$ with $X$ and other features, equation (15) can be transformed into equation (17) according to 3 as follows:

$$E[X|\mathrm{do}(Y_1), \mathrm{do}(Y_3), \ldots, \mathrm{do}(Y_{n-1})] = \sum_x \sum_{y_2} \ldots \sum_{y_n} xP(x|y_2, \ldots, y_n)P(y_2)\ldots P(y_n). \tag{17}$$

If equation (17) holds, then the causal relationship in the case can be recovered based on the factual causal direction between cyberattacks and anomalous features, as shown in Figure 8.

According to equation (17), if intervention is made on $Y_1, Y_3, \ldots, Y_{n-1}$, then the intensity of causal effect between $Y_1, Y_3, \ldots, Y_{n-1}$ and $X_k$ is

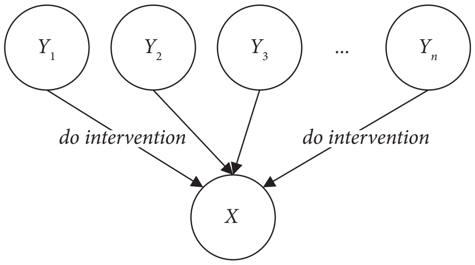


Figure 7: Intervention process.

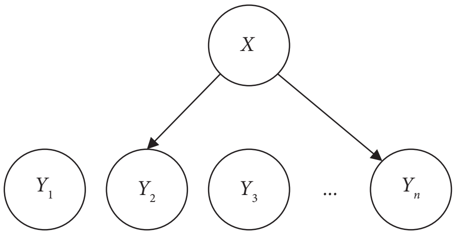


Figure 8: A factual causal relationship between a single cyberattack and features.

$$\xi_k = \sum_L \left| E[Y_i = 1|\mathrm{do}(X_k = 1)] - E[Y_i = 1|\mathrm{do}(X_k = 0)] \right|, \tag{18}$$

where $L$ is 1, 3, …, $N$−1 if $\xi_1, \xi_2, \ldots, \xi_n \leq \delta$; thus, the BN of cyberattack and features can be simplified (Figure 9).

As displayed in Figure 9, features $y_1$, $y_3$, and $y_{n-1}$ can be deleted when data are preprocessed according to the abovementioned method, and the causality is simplified as

$$\left\{\begin{matrix} X_1 \\ \vdots \\ X_k \end{matrix}\right\} \longrightarrow \left\{\begin{matrix} y_{11}, y_{12}, y_{13}, \ldots, y_{1n} \\ \vdots \\ y_{k1}, y_{k2}, y_{k3}, \ldots, y_{kn} \end{matrix}\right\} \Rightarrow \left\{\begin{matrix} X_1 \\ \vdots \\ X_k \end{matrix}\right\} \longrightarrow \left\{\begin{matrix} y_{12}, y_{14}, \ldots, y_{1n} \\ \vdots \\ y_{k2}, y_{k4}, \ldots, y_{kn} \end{matrix}\right\}. \tag{19}$$

*4.3.3. The Process of Feature Selection.* Based on the above method, all noise features satisfying Definition 2 will be deleted. Only the causal features are retained, and the selection process is as shown in Algorithm 1.

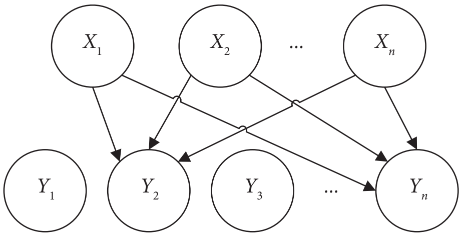


Figure 9: Factual causal relationships between multiple cyberattacks and features.

```
Input: $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_\mathbf{N}\}$, and set $P$ represents the features set, which contains $N$ features
Output: $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_{\mathbf{Cn}} | \mathbf{Cn} \leq \mathbf{N}\}$, and $\mathbf{C}$ is a causal feature set, which contains $\mathbf{Cn}$ features
Ctmax = 0 // Ctmax represents the maximum set of deleted features
Cu[N] = ∅ // Cu[i] represents the set of features that have been deleted from the $i_{th}$ feature in Set $P$
for $i$ from $1$ to $N$
    for $j$ from $i$ to $N+i-1$
        $\mathbf{E}_{\mathbf{j\%N}} = \mathbf{E}[\mathbf{X}|\mathbf{Y}_{\mathbf{j\%N}} = 1] - \mathbf{E}[\mathbf{X}|\mathbf{do}(\mathbf{Y}_{\mathbf{j\%N}} = 0)]$
        if $\mathbf{E}_{\mathbf{j\%N}} \leq \mathbf{N} * \boldsymbol{\delta}$
            Delete the j%N feature
            Cu[i] ∪ j //Noise features numbers are stored in Cu sets
        end if
    end for
end for
Count = []; it represents a set of noise features
for $i$ from $1$ to $N$//. Compare the set of features of all Cun[i] and assign the set with the most noise features set to Count
    if len(Count) < len(cun[i])
        then count = cun[i]
    end if
end for
for $i$ from 0 to len (Count)
    Delete all noise features in the Cun[i] collection;
end for
output the causal feature set $C$.
```

Algorithm 1: Causal reasoning-based feature selection (CRFS).

*4.4. Classification of Cyberattacks.* Although the causality is simplified after feature selection, as shown in Figure 9, there is still a many-to-many relationship between cyberattacks and traffic features. The key of counterfactual detection algorithm is how to choose the most appropriate labeled attacks to explain the causality of the features. According to the causal inference, it can be assumed that the possibility of changes in the results of the counterfactual detection is associated with certain changes in preconditions; thus, it can provide the basis for the causality judgment according to the magnitude of the causality. For instance, in order to quantify the causality of anomalous features caused by a cyberattack in a NIDS, the counterfactual detection can be used for inference.

As illustrated in Figure 10, the left is the fact graph, and the right is the counterfactual graph. All variables with apostrophes in the counterfactual conditions are equal to the variables without apostrophes in the fact conditions. It is assumed that, under the condition of a given evidence $\varepsilon = e$ and intervention that sets $X$ to the value of 0, the counterfactual likelihood can be calculated as $p$ $(\varepsilon = e' | \varepsilon = e$, do $(X = 0))$. Therefore, through counterfactual inquiry, a formal language can be provided to quantify the probability of a counterfactual anomalous feature $e' = 1$ when it is only assumed that the attack $X = 0$.

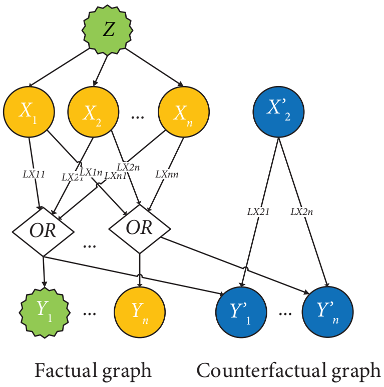


Figure 10: A twin network for counterfactual detection.

*Definition 3* (expected sufficiency [26]). The expected sufficiency of cyberattack $X_a$ is the number of anomalous features that would expect to persist if the intervention is given to switch off all other possible causes of the anomalous features:

$$E(X_a, \varepsilon) = \sum_{Y'} |Y'_+| P\left(Y' | \varepsilon, \mathrm{do}\left(pa \frac{(Y_+)}{X_a}\right) = 0\right), \tag{20}$$

where $X_a$ denotes the type of cyberattack $a$, $Y_+$ indicates the anomalous features in the fact conditions, Pa $(Y_+)$ denotes the parent node of $Y_+$ that represents all cyberattacks that may result in the anomalous feature $Y$, Pa $(Y_+)\backslash X_a$ is the parent node of $Y_+$ except for $X_a$, $Y'_+$ represents the anomalous features in counterfactual situations, and $\varepsilon$ denotes the set of all factual evidence features. If $E$ $(X_a, \varepsilon)$ is maximum in the set for all $E$ $(X, \varepsilon)$, the cyberattack type $X_a$ will be a causal explanation for the given evidence $\varepsilon$.

*Inference 1.* According to equation (19) and SCM [26, 51], the expected sufficiency of cyberattack $X_a$ is given by

$$E(X_a, \varepsilon) = \frac{1}{P(Y_\pm)} \sum_{Z \in Y_+} (-1)^Z P$$

$$(Y_- = 0, Z = 0, X_a = 1) * Q(a, Z), \tag{21}$$

$$Q(a, Z) = \sum_{Y \in Y_+/Z} \left(1 - L_{X_a, Y}\right)^Z,$$

where $Y\_$ denotes the normal feature in the set of all factual evidence features. It is mainly very complicated and cumbersome to solve noisy and exogenous variables, while it is unnecessary to solve these variables in equation (20). At the same time, the value of $L$ can be calculated based on the prior data. Therefore, equation (20) obtained through counterfactual reasoning greatly simplifies the causal relationship between cyberattacks and traffic features.

*4.5. Time Complexity.* To determine the time complexity of the proposed causal ML-based NIDS, it is required to determine the complexity of each algorithm used in each stage. As the performance of different algorithms at different stages is compared, the overall time complexity is determined by that algorithm, producing the highest overall complexity. It is assumed that the dataset is composed of $M$ samples and $N$ features. In general, $M \gg N$.

Starting with the data preprocessing stage, the complexity of the $Z$-score and Min-Max normalization is $O$ $(N)$. As it is required to normalize all the samples of the $N$ features within the dataset, the complexity of deleted incorrect and fuzzy row sets is $O$ $(M)$. Therefore, the overall complexity of the first phase is $O$ $(M)$.

The time complexity of the second stage is $O$ $(N^2)$. Firstly, this phase intervenes all the features, and only $N$ steps are taken and compared with $(N-1)/2$ features. In the third stage, the complexity of the KNN classifier can be estimated as $O$ $(M_\mathrm{l} * K)$ [9], and the time complexity of the random forest is $O$ $(M_l * K * D)$, where $K$ $(K < N)$ is the dimension after feature selection, $M_l$ denotes the number of samples after deleting the incorrect and fuzzy row sets, and $D$ is the depth of the tree. The time complexity of the fourth stage is $O$ $(T * M_l * K)$, where $T$ $(T < M$ and $T < D)$ represents the type of a cyberattack selected in the third stage.

Based on the aforementioned discussion, the overall complexity of the proposed framework is $O$ $(M_l * K * D)$. The time complexity of data preprocessing and feature selection is $O$ $(M + N^2)$. As $M \gg N$, the time complexity of data preprocessing and feature selection is approximately equal to $O$ $(M)$, and this time complexity is far less than the time complexity $O$ $(M * N^2)$ of feature selection, including MOMBNF [9]. Finding the overall time complexity is highly critical because the model will often be retrained to learn new patterns of cyberattacks.

## 5. Performance Evaluation

*5.1. Experimental Setting.* The CICIDS19 dataset was launched in 2019 by the Canadian Institute for Cybersecurity, and it contains benign and the most up-to-date common cyberattacks, which is similar to real-world data with a total of 87 features [47]. This dataset contains 11 types of attacks: DRDOS_MSSQL, DRDOS_SNMP, SYN, DRDOS_NTP, TFTP, UDP-LAG, DRDOS_NETBIOS, DRDOS_DNS, DRDOS_UDP, DRDOS_LDAP, and DRDOS_SSDP. As shown in Table 1, it also includes the results of network traffic features based on timestamps, source and target IPs, source and target ports, protocols, and attack token flows.

The raw network packet for UNSW_NB15 [52] was created by the Australian Cyber Security Center, and it is a comprehensive set of cyberattack traffic data. Compared with other datasets, these two datasets are more appropriate for the research on NIDSs. UNSW_NB15 dataset has nine types of cyberattacks, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. As presented in Table 2, tools, such as Argus, are used by UNSW-NB15 to generate a total of 49 features with similar labels.

NSL-KDD [53, 54] contains 7 major categories of attacks, such as ipsweep, Neptune, nmap, portsweep, Satan, smurf, and teardrop. NSL-KDD elimination of redundant records in the training set helps classifiers to be unbiased toward more frequent records. The training and test sets contain a reasonable number of instances, which can be used as a valid benchmark dataset to help researchers compare different intrusion detection methods. As shown in Table 3, there are 41 dimensional features in NSL-KDD.

The fuzzy logic system (FLS) [47] is used to evaluate the quality of realism of CICIDS19, UNSW-NB15, and NSL-KDD datasets. The FLS is based on Sugeno fuzzy model [55] that investigates the quality of realism of IDS dataset. The CICIDS19, UNSW-NB15, and NSL-KDD datasets contain a set of network intrusion attacks that reflect real-world standards. The generation process fully considers the characteristics of network intrusion attacks and the dynamics of the network.

Table 1: Description of features in the CICIDS19 dataset.

| Feature name | Description |
|---|---|
| Flow ID | Flow ID |
| S IP | Source Ip |
| S Port | Source port number |
| D IP | Destination IP |
| D Port | Destination port |
| Protocol | Representation number of the protocol |
| Timestamp | Timestamp |
| Flow duration | Duration of the flow in microsecond |
| TFwd packets | Total packets in the forward direction |
| TB packets | Total packets in the backward direction |
| TL of Fwd packets | Total size of packet in forward direction |
| TL of Bwd packets | Total size of packet in backward direction |
| Fwd PL max | Maximum size of packet in forward direction |
| Fwd PL min | Minimum size of packet in forward direction |

Table 2: Description of features in the UNSW_NB15 dataset.

| Feature name | Description |
|---|---|
| Flow ID | Flow id |
| Srcip | Source Ip |
| Sport | Source port number |
| Dstip | Destination IP address |
| Dsport | Destination port number |
| Proto | Representation number of the protocol |
| Dur | Record total duration |
| Spkts | Source to destination packet count |
| Dpkts | Destination to source packet count |
| Sjit | Source jitter (mSec) |
| Sintpkt | Source interpacket arrival time (mSec) |
| Ct_ftp_cmd | No of flows that has a command in ftp session |
| Tcprtt | The sum of "synack" and "ackdat" of the TCP |
| Ltime | Record last time |

In order to use a variety of algorithms more effectively, python was used to implement our model. The hardware and software specifications are summarized in Table 4.

*5.2. The Results of Experiments.* This section presents three sets of experiments to verify the effectiveness of the proposed causal ML-based NIDS.

*5.2.1. Influences of Data Preprocessing on the Training Samples.* Concerning the effects of data preprocessing on the size of training samples, the learning curve of training accuracy and cross-validation accuracy with the change of the size of training samples could be obtained. Because the amount of data in the datasets is large enough, about 10% of the data can be used as the test set to work well, so a 90:10 split is used for normalization in this paper. After normalization, using the 90/10% splitting criteria, the two datasets are randomly divided into training and test datasets.

Table 3: Description of features in the NSL-KDD dataset.

| Feature name | Description |
|---|---|
| Protocol type | Type of protocol (TCP, UDP...) |
| Source bytes | No. of B from source to destination |
| Wrong fragments | No. of wrong fragments |
| Urgent | No. of urgent packets |
| Error rate | % of connections with SYN errors |
| Failed logins | No. of unsuccessful attempts at login |
| Logged in | If logged in, 1/if login failed, 0 |
| Same srv rate | % of connections to the same service |
| Count | No. of connections to the same host as the current connection at a given interval |
| Dst host srv rate | % of connections to different hosts on the same system |
| # Root | No. of root accesses |
| # Shells | No. of active command interpreters |
| Dst host srv serror rate | % of connections to a host and specified service with an S0 error |

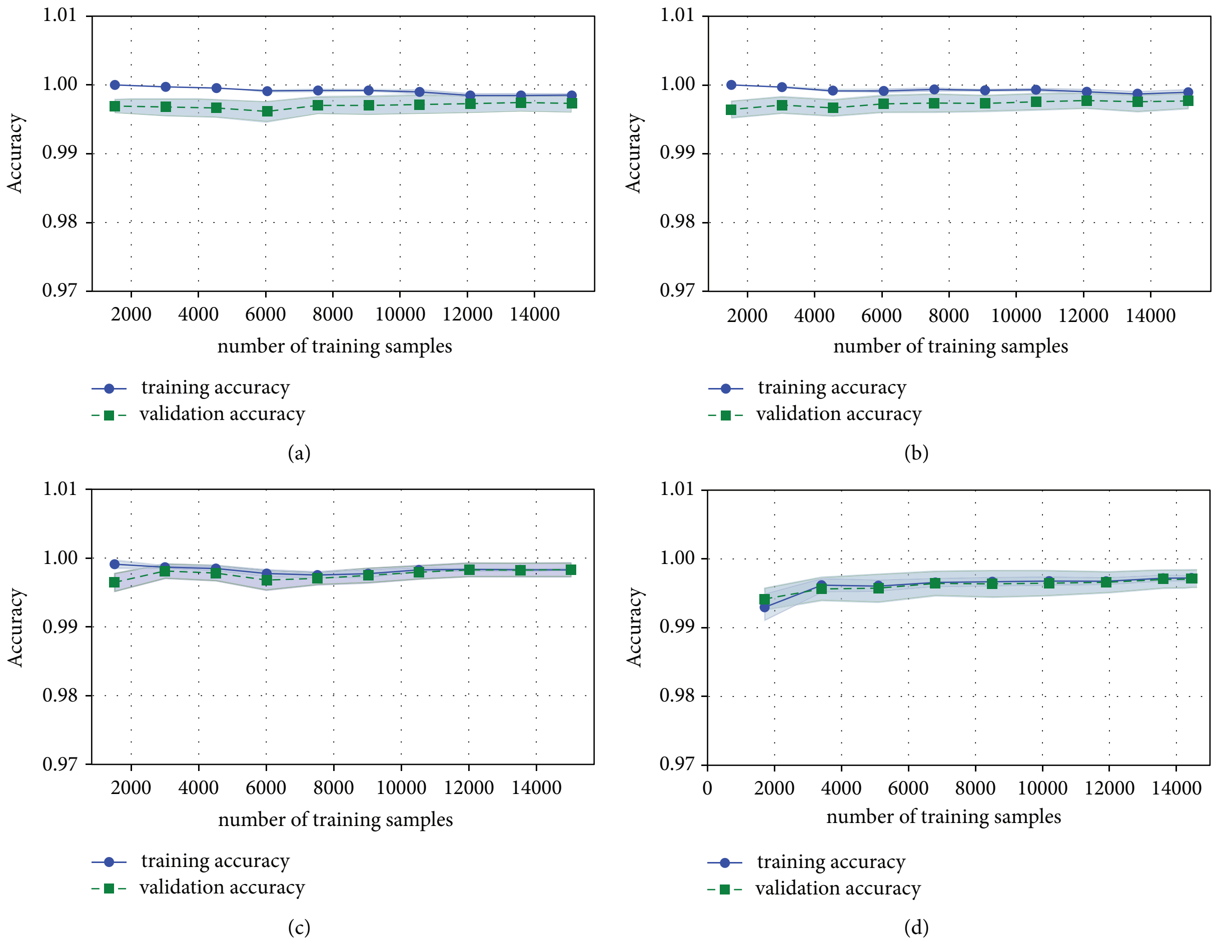
Table 4: The software and hardware specifications.

| Hardware specifications | Software specifications |
|---|---|
| Processor: Intel (R) Core (TM) i5-8265U CPU | Operating system: Windows 10 |
| Memory: 8.00 GB | Programming language: Python 3.8<br>Development tools: PyCharm 2019 |
| Graphics card: NVIDIA Geforce MX250 | Packages: pandas, numpy, sklearn |

*(1) Influences of Data Preprocessing on the Size of Training Samples.* In this study, *Z*-score, SMOTE [56–58], CFS [9, 59–61], and CRFS (causal reasoning-based feature selection) were used for making comparison. The SMOTE algorithm is used for SMOTE to sample a few classes after data processing by the *Z*-score, and CFS selects features after data processing by the SMOTE. For the CRFS method proposed in this paper, the causal reasoning-based feature selection presented in Section 4.3 is used after data processing by the *Z*-score. The cross-validation curves of different datasets under different types of cyberattacks after data processing by the four methods mentioned above are shown in Figures 11-12.

Figure 11 compares the accuracy with the number of training samples required for the four methods (it is considered that there is only one type of cyberattack here, all cyberattacks are treated as one type of cyberattack, and its name is "abnormal"). As depicted in Figure 11, to converge the training accuracy and cross-validation accuracy, the number of training samples required for the *Z*-score and SMOTE was more than 16,000, which was within 10,000 for the CFS; however, the number of training samples required for the CRFS was only about 5,000, which was significantly lower than that of the *Z*-score, SMOTE, and CFS, while it could ensure the same training accuracy.

The accuracy and the number of training samples required for the four methods (it is considered that there are multiple types of cyberattacks here) were compared (Figure 12). As shown in Figure 12, in order to converge the training accuracy and cross-validation accuracy, the number of training samples required for the Z-score and SMOTE was



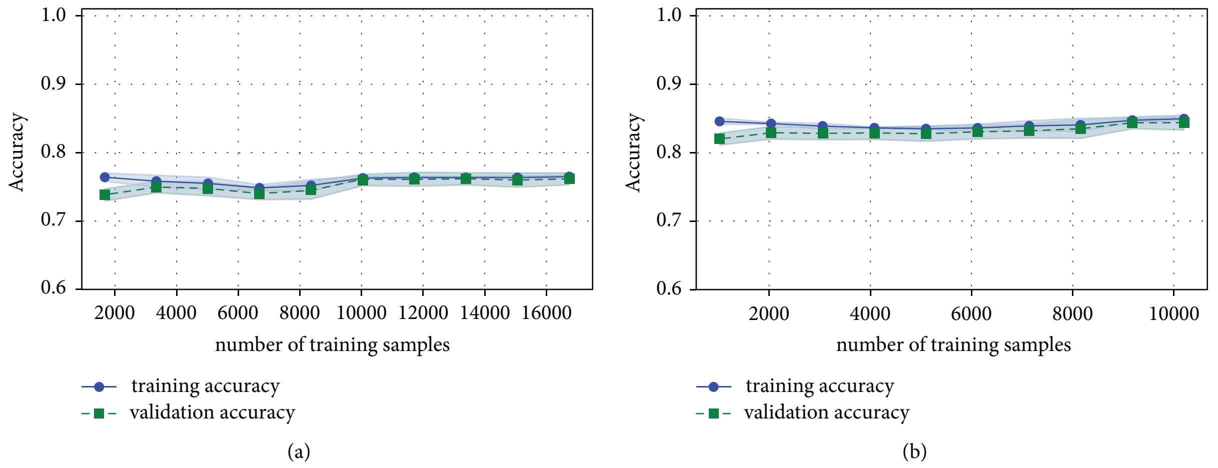
(a)

(b)

(c)

(d)

FIGURE 11: Learning curves comparing the accuracy with the number of training samples required for the four methods (single cyberattack). (a) *Z*-score. (b) SMOTE. (c) CFS. (d) CRFS.



(a)

(b)

FIGURE 12: Continued.

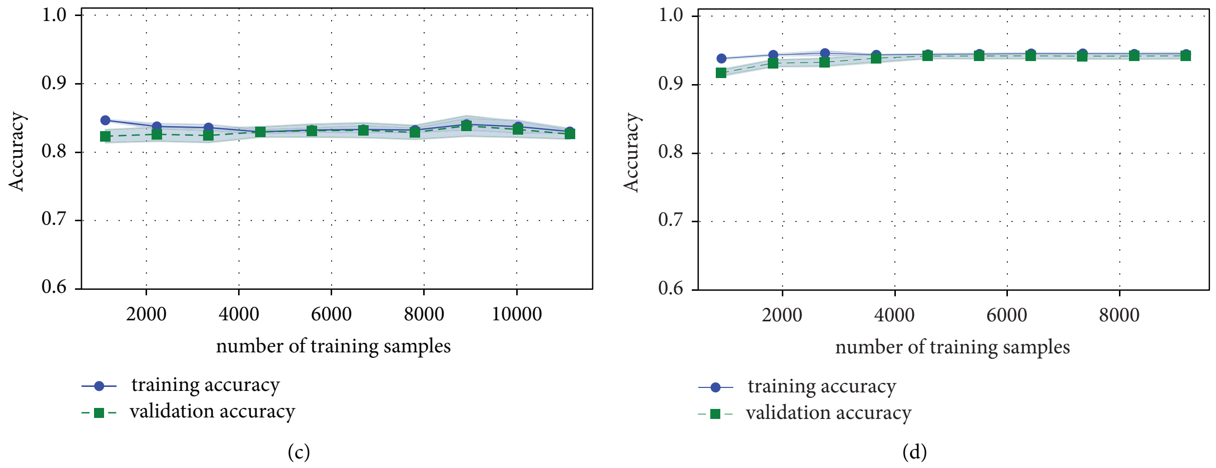


Figure 12: The accuracy and the number of training samples required for the four methods (there are multiple types of cyberattacks). (a) *Z*-score. (b) SMOTE. (c) CFS. (d) CRFS.

close to 10,000. The number of training samples required for the CFS was within 5,000, and the number of training samples required for the CRFS was close to 4,000, which decreased by 60%, 60%, and 20% compared with those of the *Z*-score, SMOTE, and CFS, respectively. Meanwhile, the training accuracy reached the highest, which significantly improved by about 10% compared with the highest training accuracy achieved by the SMOTE.

As illustrated in Figures 11 and 12, with the increase of types of cyberattacks, the number of training samples required for the *Z*-score, SMOTE, and CFS significantly increased, while the training accuracy noticeably decreased. As for the number of training samples required for the CRFS, it basically remained below 5,000 samples and the training accuracy slightly decreased. This highlights the positive influence of utilizing the CRFS technique, as it could significantly reduce the size of the required training samples without sacrificing the detection performance.

*(2) Influences of Data Preprocessing on the Time Required for Training.* To further highlight the influences of the data preprocessing stage, Table 5 summarizes the time required for different methods to construct the learning curve under different types of cyberattacks. For instance, when there were two types of cyberattacks, nearly 483 s was needed for the Z-score to establish the learning curve, which was reduced to 370 s after processing by the SMOTE and 154 s after processing by the CFS. However, the time required to construct the learning curve after processing for the CRFS was only 90 s, which was 81.4%, 75.7%, and 41.6% lower than that of the Z-score, SMOTE, and CFS, respectively.

This indicates that CRFS can not only guarantee the accuracy of detection but also effectively reduce the time required for training. The proof mentioned in Section 4.5 verifies that the feature selection algorithm proposed in this article has lower time complexity than the other algorithms. As the noisy features are deleted by the CRFS, the ML algorithms only need to fit causal features. The accuracy of the subsequent steps can be guaranteed and the time complexity required for training can be reduced.

Table 5: Time required to construct the learning curve.

| Number of types of cyberattacks | Algorithm | | | |
|---|---|---|---|---|
| | *Z*-score | SMOTE | CFS | CRFS |
| 1 | 483 | 370 | 154 | 90 |
| 11 | 679 | 671 | 431 | 314 |

*5.2.2. Influences of Feature Selection Methods on the Number of Features Required.* In this experiment, three groups of control experiments were set, and the number of features and the training accuracy after data processing by the SMOTE, CFS, and Min-Max were compared. The CRFS algorithm was used to further select features. SMOTE, CFS, and Min-Max add (do) in Tables 6–17 indicated that the CRFS method could be applied to process and select the data after the data processing by these methods.

The number of features left after processing by different algorithms in the CICIDS19 dataset under different types of cyberattacks is shown in Table 6. After processing by the CRFS algorithm, the number of features required for training was decreased by more than 50% at the minimum and 94% at the maximum compared with that before processing. Moreover, the number of features processed by the CRFS algorithm was significantly lower than that calculated by the CFS algorithm. This may be related to the fact that CRFS based on causal reasoning only selects network features that have a causal relationship with the cyberattacks, and it eliminates the features with a spurious correlation. The CFS is a feature selection method based on high correlation, which can greatly reduce the number of features. However, this method also selects some noncausal features with a spurious correlation, resulting in the higher number of features than that of CRFS.

The detection accuracy between SMOTE and CRFS, between CFS and CRFS, and between min-max and CRFS in the CICIDS19 dataset was, respectively, shown in Tables 7–9. As presented in the abovementioned tables, although the

Table 6: The number of features selected by different feature selection methods in CICIDS19 dataset.

| The number of attacks | Feature selection method (the number of features required) | | | | | |
|---|---|---|---|---|---|---|
| | SMOTE | SMOTE (do) | CFS | CFS (do) | Min-max | Min-max (do) |
| 1 | 70 | 5 | 9 | 4 | 70 | 4 |
| 3 | 70 | 6 | 8 | 3 | 70 | 5 |
| 7 | 70 | 6 | 13 | 6 | 70 | 5 |
| 11 | 70 | 8 | 16 | 7 | 70 | 14 |

Table 7: Comparison of accuracy between SMOTE and CRFS in the CICIDS19 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 3, 7, 11) | | | |
|---|---|---|---|---|
| | 1 | 3 | 7 | 11 |
| SMOTE | 0.9995 | 0.9962 | 0.9109 | 0.8758 |
| SMOTE (do) | 0.9995 | 0.9894 | 0.9042 | 0.8716 |
| Percentage | 1 | 0.9932 | 0.9926 | 0.9952 |

Table 8: Comparison of accuracy between CFS and CRFS in the CICIDS19 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 3, 7, 11) | | | |
|---|---|---|---|---|
| | 1 | 3 | 7 | 11 |
| KNN-CFS | 0.9985 | 0.9953 | 0.9735 | 0.8917 |
| KNN-CFS (do) | 0.9981 | 0.9953 | 0.9715 | 0.8895 |
| Percentage | 0.9995 | 1 | 0.9979 | 0.9975 |

Table 9: Comparison of accuracy between min-max and CRFS in the CICIDS19 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 3, 7, 11) | | | |
|---|---|---|---|---|
| | 1 | 3 | 7 | 11 |
| Min-max | 0.9995 | 0.9953 | 0.8470 | 0.8420 |
| Min-max (do) | 0.9995 | 0.9891 | 0.8359 | 0.8302 |
| Percentage | 1 | 0.9938 | 0.9869 | 0.9860 |

Table 10: The number of features extracted by different feature selection methods in the UNSW-NB15 dataset.

| Number of types of cyberattacks | Feature selection method (number of features required) | | | | | |
|---|---|---|---|---|---|---|
| | SMOTE | SMOTE (do) | CFS | CFS (do) | Min-max | Min-max (do) |
| 1 | 40 | 7 | 6 | 6 | 40 | 19 |
| 9 | 40 | 7 | 11 | 5 | 40 | 20 |

Table 11: Comparison of accuracy between SMOTE and CRFS in the UNSW-NB15 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 9) | |
|---|---|---|
| | 1 | 9 |
| SMOTE | 0.9357 | 0.8147 |
| SMOTE (do) | 0.9337 | 0.7499 |
| Percentage | 0.9979 | 0.9205 |

Table 12: Comparison of accuracy between CFS and CRFS in the UNSW-NB15 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 9) | |
|---|---|---|
| | 1 | 9 |
| KNN-CFS | 0.9213 | 0.7869 |
| KNN-CFS (do) | 0.9213 | 0.7326 |
| Percentage | 1 | 0.931 |

Table 13: Comparison of accuracy between min-max and CRFS in the UNSW-NB15 dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 9) | |
|---|---|---|
| | 1 | 9 |
| Min-max | 0.9435 | 0.8496 |
| Min-max (do) | 0.9455 | 0.8448 |
| Percentage | 1.002 | 0.9944 |

Table 14: The number of features extracted by different feature selection methods in the NSL-KDD dataset.

| The number of attacks | Feature selection method (the number of features required) | | | | | |
|---|---|---|---|---|---|---|
| | SMOTE | SMOTE (do) | CFS | CFS (do) | Min-max | Min-max (do) |
| 1 | 36 | 8 | 8 | 7 | 36 | 10 |
| 7 | 36 | 8 | 12 | 7 | 36 | 10 |

Table 15: Comparison of accuracy between SMOTE and CRFS in the NSL-KDD dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 7) | |
|---|---|---|
| | 1 | 7 |
| SMOTE | 0.9951 | 0.9714 |
| SMOTE (do) | 0.9907 | 0.9701 |
| Percentage | 0.9956 | 0.9987 |

Table 16: Comparison of accuracy between CFS and CRFS in the NSL-KDD dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 7) | |
|---|---|---|
| | 1 | 7 |
| KNN-CFS | 0.9835 | 0.9681 |
| KNN-CFS (do) | 0.9835 | 0.9624 |
| Percentage | 1 | 0.9960 |

Table 17: Comparison of accuracy between min-max and CRFS in the NSL-KDD dataset.

| Feature selection method | Accuracy under different types of cyberattacks (1, 7) | |
|---|---|---|
| | 1 | 7 |
| Min-max | 0.9971 | 0.9751 |
| Min-max (do) | 0.9979 | 0.9748 |
| Percentage | 1.0008 | 0.9997 |

Table 18: Performance of different classifiers in CICIDS19 dataset under different types of cyberattacks.

| Algorithm | Detection accuracy under different types of cyberattacks (1, 3, 7, 11) | | | |
|---|---|---|---|---|
| | 1 | 3 | 7 | 11 |
| RS-KNN-CFS | 0.9985 | 0.9953 | 0.9735 | 0.8917 |
| TPE-KNN-CFS | 0.9954 | 0.9942 | 0.9687 | 0.8793 |
| RS-KNN-IGBS | 0.9938 | 0.4577 | 0.4157 | 0.2887 |
| TPE-KNN-IGBS | 0.9864 | 0.3633 | 0.3024 | 0.2697 |
| RS-RF-CFS | 0.9986 | 0.9948 | 0.9676 | 0.8951 |
| TPE-RF-CFS | 0.9987 | 0.9947 | 0.9529 | 0.8921 |
| RS-RF-IGBS | 0.9928 | 0.4561 | 0.4170 | 0.2963 |
| TPE-RF-IGBS | 0.9883 | 0.4534 | 0.4033 | 0.2965 |
| BRS | 0.9985 | 0.9461 | 0.7869 | 0.7732 |
| CMLN | 0.9995 | 0.9993 | 0.9856 | 0.9852 |

Table 19: Performance of different classifiers in UNSW-NB15 dataset under different types of cyberattacks.

| Algorithm | Detection accuracy under different types of cyberattacks (1, 9) | |
|---|---|---|
| | 1 | 9 |
| RS-KNN-CFS | 0.9283 | 0.7869 |
| TPE-KNN-CFS | 0.9168 | 0.7654 |
| RS-KNN-IGBS | 0.9501 | 0.7869 |
| TPE-KNN-IGBS | 0.9450 | 0.7073 |
| RS-RF-CFS | 0.9209 | 0.7806 |
| TPE-RF-CFS | 0.9274 | 0.7915 |
| RS-RF-IGBS | 0.9198 | 0.7253 |
| TPE-RF-IGBS | 0.9198 | 0.7367 |
| BRS | 0.8717 | 0.8082 |
| CMLN | 0.9926 | 0.9229 |

Table 20: Performance of different classifiers in NSL-KDD dataset under different types of cyberattacks.

| Algorithm | Detection accuracy under different types of cyberattacks (1, 7) | |
|---|---|---|
| | 1 | 7 |
| RS-KNN-CFS | 0.9886 | 0.9795 |
| TPE-KNN-CFS | 0.9850 | 0.9778 |
| RS-KNN-IGBS | 0.9911 | 0.9797 |
| TPE-KNN-IGBS | 0.9919 | 0.9812 |
| RS-RF-CFS | 0.9877 | 0.9671 |
| TPE-RF-CFS | 0.9837 | 0.9698 |
| RS-RF-IGBS | 0.9939 | 0.9821 |
| TPE-RF-IGBS | 0.9923 | 0.9810 |
| BRS | 0.9959 | 0.9538 |
| CMLN | 0.9983 | 0.9933 |

number of features required for training was markedly reduced after data processing by the CRFS algorithm, its training accuracy still maintained about 99% of the original algorithm's accuracy, and the decrease could be almost negligible compared with the number of compressed features. The results showed that the CRFS algorithm could not only effectively reduce the number of training samples required for processing but also ensure the accuracy of training samples to a relatively stable level. This is because the CRFS algorithm can identify the real causal relationship between cyberattacks and features, while the eliminated features are only features of spurious correlation, slightly influencing accuracy.

The number of features left in the UNSW-NB15 dataset after data processing by different algorithms under different types of cyberattacks is shown in Table 10. After further processing of features by the CRFS algorithm, the minimum and maximum reduction of the number of features required for training was >50% and >82.5% compared with that before processing. When there were few types of cyberattacks, the effect of applying causality to the data processed by the CFS to find compressed features was significantly reduced. Owing to the strong correlation and strong causality, UNSW-NB15 was consistent after the data processing by the CFS. However, when there were several types of cyberattacks, the reduction was also significant, up to 54.5%, after further processing by the CRFS algorithm.

The detection accuracy between SMOTE and CRFS, between CFS and CRFS, and between min-max and CRFS in the UNSW-NB15 dataset was, respectively, shown in Tables 11–13. As presented in the abovementioned tables, when there were few types of cyberattacks, although the number of features required for training was noticeably reduced after processing by the CRFS algorithm, the accuracy of training basically remained unchanged and the effect was obvious.

In the NSL-KDD dataset, after further processing of features by the CRFS algorithm, the maximum reduction of the number of features required for training was >82.5%. As presented in the abovementioned dataset, the number of features required for training was noticeably reduced after processing by the CRFS algorithm in the NSL-KDD dataset.

To sum up, the CRFS algorithm could effectively reduce the number of required training samples in the CICIDS19, UNSW-NB15, and NSL-KDD datasets, and it could also ensure the accuracy of training samples with a relatively acceptable stability. Especially, under the circumstance of a smaller number of cyberattacks, with a greatly reduced complexity in time and calculation, the training accuracy was basically unchanged. It was proved that causal features could not only complete the NIDS detection task but also ensure the stability of the accuracy rate. The selected causal features might provide a targeted help for the next preventive treatment.

*5.2.3. Influences of Different Types of Cyberattacks on the Detection Performance.* To evaluate the performance of the different classifiers and study the effects of the different optimization methods, it can be referred to the evaluation index of accuracy of test data (ACC). Random search (RS) and tree-structured Parzen estimator (TPE) are two optimal parameter adjustment methods with the highest accuracy of the KNN and random forest in MOMBNF [9]. CMLN is a causal ML-based NIDS.

Performance of different classifiers in CICIDS19, UNSW-NB15, and NSL-KDD datasets under different types of cyberattacks was compared in Tables 18–20. As shown in Table 18, in the CICIDS19 dataset, with an increase in the types of cyberattacks, the detection accuracy in MOMBNF significantly decreased. When there were 11 types of cyberattacks, the detection accuracy of all the parameter optimization methods in MOMBNF was lower than 90%, especially the accuracy of the test set was lower than 30% after IGBS data processing. However, after CMLN training, the accuracy of the test set was stable at more than 98.5%, which was about 9% higher than the optimal RS-KNN-CFS method. It can be seen from Tables 18–20 that, regardless of the composition of the datasets, the accuracy of CMLN test set was higher than that of MOMBNF and BRS [47], especially when there were several types of cyberattacks. The detection rate of CMLN was higher than that of MOMBNF.

## 6. Conclusions

Although ML aims to facilitate the detection of anomalies, it is important to first understand how detection is performed and clearly define the desired output of our algorithms. When traditional ML algorithms cannot decouple correlation and causality, it is difficult to achieve a stable prediction [8]. Therefore, this paper proposed a novel causal ML-based NIDS. Firstly, by establishing a causal link between cyberattacks and features through causal intervention, the noisy features could be deleted and the minimum size of training features could be determined. Then, the ML and counterfactual detection algorithm were used to find out the unique label. Finally, CICIDS19, UNSW-NB15, and NSL-KDD datasets were utilized to evaluate the performance of the proposed detection method.

The results of experiments showed that the CRFS method proposed in this paper could reduce the size of training samples and training time by at least 40%. Meanwhile, the number of features required for training was greatly reduced after data processing by the CRFS algorithm, and it also ensured the accuracy of training with a relatively acceptable stability. It was proved that the deletion of noisy features did not affect the accuracy of detection. The results showed that compared with other optimization techniques, CMLN has the highest detection accuracy (when there were 11 types of cyberattacks, the accuracy was improved by nearly 9% compared with the optimal RS-KNN-CFS method). It was confirmed that the counterfactual detection algorithm could effectively identify the causal relationship between features and the type of cyberattacks.

At present, new cybersecurity threats are becoming ever severe, which cannot be classified according to the existing classification methods. Hence, how to effectively combine unsupervised learning and causal ML to construct new NIDs to detect new cybersecurity threats may be a new direction for investigation.

## Data Availability

The data used to support the findings of this study can be accessed from https://www.unb.ca/cic/datasets/index.html, https://ieee-dataport.org/documents/unswnb15-dataset, and https://www.unb.ca/cic/datasets/nsl.html.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by Natural Science Foundation of China (no. 61972412) and National Key Research and Development Program of China (no. 2018YFB0204301).

## References

[1] A. Kozowski, “Comparative analysis of cyberattacks on Estonia, Georgia and Kyrgyzstan,” *European Scientific Journal*, vol. 3, 2020.

[2] M. Li, D. Han, X. Yin et al., “Design and implementation of an anomaly network traffic detection model integrating temporal and spatial features,” *Security and Communication Networks*, vol. 2021, Article ID 7045823, 15 pages, 2021.

[3] S. Moualla, K. Khorzom, and A. Jafar, “Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 5557577, 13 pages, 2021.

[4] E. K. Viegas, A. O. Santin, and V. Abreu, “Machine learning intrusion detection in big data era: a multi-objective approach for longer model lifespans,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, 2020.

[5] Z. Kamil, R. Yusof, N. Bahaman, M. A. Salama, and F. M. F. Cik, “Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset,” *IEEE Access*, vol. 9, 2017.

[6] E. Tsukerman, *Designing a MACHINE LEARNING Intrusion Detection System: Defend Your Network from Cybersecurity Threats*Apress, NY, USA, 2020.

[7] J. Zhang and M. Zulkernine, “Anomaly based network intrusion detection with unsupervised outlier detection,” in *Proceedings of the 2006 ICC06. IEEE International Conferenceon Communications,* vol. 5, pp. 2388–2393, IEEE, Istanbul, Turkey, June 2006.

[8] K. Kuang, B. Li, P. Cui, J. Tao, F. Zhuang, and F. Wu, “Stable Prediction Via Leveraging Seed Variable,” 2020, https://arxiv.org/abs/2006.05076.

[9] M. N. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “Multi-stage optimized machine learning framework for network intrusion detection,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, 2020.

[10] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid kpca and svm with ga model for intrusion detection," *Applied Soft Computing*, vol. 18, pp. 178–184, 2014.

[11] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670–2679, 2015.

[12] M. Wazid and A. K. Das, "An efficient hybrid anomaly detection scheme using K-means clustering for wireless sensor networks," *Wireless Personal Communications*, vol. 90, no. 4, pp. 1971–2000, 2016.

[13] A. Moubayed, M. Injadat, A. B. Nassif, L. Hanan, and A. Shami, "Elearning: challenges and research opportunities using Machine Learning data analytics," *IEEE Access*, vol. 6, pp. 117–139, 2018.

[14] S. Aliawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, 2017.

[15] X. Sun, J. Dai, P. Liu, N. Anoop, and S. John, "Using bayesian networks for probabilistic identification of zero-day attack paths," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, 2018.

[16] T. A. Tang, L. Mhamdi, and D. Mclernon, "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking," in *Proceedings of the International Conference on Wireless Networks & Mobile Communications*, IEEE, Fez, Morocco, October 2016.

[17] A. A. Daya, M. A. Salahuddin, N. Limam, and R. Boutaba, "BotChase: graph-based bot detection using machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 15–29, 2020.

[18] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, Article ID 82512, 2019.

[19] A. Mahindru and A. Sangal, "SemiDroid: a behavioral malware detector based on unsupervised machine learning techniques using feature selection approaches," *International Journal of Machine Learning and Cybernetics*, vol. 12, 2021.

[20] H. Liu and Z. Zhao, "Manipulating Data And Dimension Reduction Methods: Feature Selection," in *Computational Complexity*, R. Meyers, Ed., Springer, NY, USA, 2012.

[21] H. Liu, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1998.

[22] X. Zhao, J. Ruan, and H. Tang, "Multi-compositional MRI evaluation of repair cartilage in knee osteoarthritis with treatment of allogeneic human adipose-derived mesenchymal progenitor cells," *Stem Cell Research & Therapy*, vol. 10, no. 1, 2019.

[23] A. Hadeel, S. Ahmad, and S. K. Eddin, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Systems with Applications*, vol. 148, Article ID 113249, 2020.

[24] A. E. Sabry, Z. Orman, and M. A. Adnan, "A new feature selection model based on id3 and bee's algorithm for intrusion detection system," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, no. 2, 2015.

[25] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.

[26] J. G. Richens, M. Ciaran, and S. J. Lee, "Improving the accuracy of medical diagnosis and causal Machine Learning," *Nature Communications*, vol. 11, no. 1, 2020.

[27] Z. J. Lim, S. K. Goh, and I. Dhief, "Causal effects of landing parameters on runway occupancy time using causal machine learning models," in *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, Australia, December 2020.

[28] A. Gelman, "Causality and statistical learning," *American Journal of Sociology*, vol. 117, no. 3, pp. 955–966, 2011.

[29] J. Pearl, "Causal inference in statistics: an overview," *Statistics Surveys*, vol. 3, pp. 96–146, 2009.

[30] L. Yao, "A survey on the causal inference," 2020, https://arxiv.org/abs/2002.02770.

[31] B. Schölkopf, "Causality for machine learning," 2019, https://arxiv.org/abs/1911.10500.

[32] B. Schlkopf, F. Locatello, S. Bauer et al., "Towards causal representation learning," 2021, https://arxiv.org/abs/2102.11107.

[33] M. Wang, C. Liu, and G. Zhi, *Statistical Approaches for Causal Inference*, Scientia Sinica Mathematica, 2018.

[34] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference (judea pearl)," *Artificial Intelligence*, vol. 48, no. 8, pp. 117–124, 1990.

[35] I. Shrier, R. W. Platt, R. J. Steele et al., "Estimating causal effects of treatment in a randomized trial when some participants only partially adhere," *Epidemiology*, vol. 29, no. 1, 2017.

[36] S. Zhu, I. Ng, and Z. Chen, "Causal and causal discovery with reinforcement learning," 2019, https://arxiv.org/abs/1906.04477.

[37] J. M. Robins, T. S. Richardson, and I. Shpitser, "An interventionist approach to mediation analysis," 2020, https://arxiv.org/abs/2008.06019.

[38] F. H. Messerli, "Chocolate consumption, cognitive function, and Nobel laureates," *New England Journal of Medicine*, vol. 367, no. 16, pp. 1562–1564, 2012.

[39] S. Chockalingam, W. Pieters, A. Teixeira, and G. Van, "Bayesian network models in cyber security: a systematic review," in *Proceedings of the The 22nd Nordic Conference on Secure IT Systems (NordSec 2017)*, pp. 105–122, Springer, Tartu, Estonia, November 2017.

[40] D. Angus and S. Peake, "Early goal-directed therapy in the treatment of sepsis: response to comments by Jaehne et al," *Intensive Care Medicine, 41*, pp. 1729-1730, 2015.

[41] V. Didelez, I. Pigeot, and P. Judea, "Causality: models, reasoning, and inference," *Politische Vierteljahresschrift*, vol. 42, no. 2, pp. 313–315, 2001.

[42] S. G. West and T. Koch, "Restoring causal analysis to structural equation modeling review of causality: models, reasoning, and inference, by Judea Pearl," *Structural Equation Modeling A Multidisciplinary Journal*, vol. 21, no. 1, pp. 484–498, 2014.

[43] D. Geiger, T. Verma, and J. Pearl, "D-separation: from theorems to algorithms," *Machine Intelligence and Pattern Recognition*, North-Holland, vol. 10, pp. 139–148, 1990.

[44] M. Khattab, "Ali alheeti, klaus McDonald-maier, "intelligent intrusion detection in external communication systems for autonomous vehicles," *Systems Science & Control Engineering*, vol. 6, no. 1, 2018.

[45] Z. Zhang, Y. Cheng, and N. C. Liu, "Comparison of the effect of mean-based method and z-score for field normalization of citations at the level of Web of Science subject categories," *Scientometrics*, vol. 101, no. 3, pp. 1679–1693, 2014.

[46] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015, https://arxiv.org/abs/1503.06462.

[47] M. Prasad, S. Tripathi, and K. Dahal, “An efficient feature selection-based Bayesian and Rough set approach for intrusion detection,” *Applied Soft Computing*, vol. 87, Article ID 105980, 2020.

[48] R. R. Tucci, “Introduction to judea pearl’s do-calculus,” 2013, https://arxiv.org/abs/1305.5506.

[49] L. Yao, Z. Chu, and S. Li, “A survey on causal inference,” 2020, https://arxiv.org/abs/2002.02770.

[50] W. Miao, C. C. Liu, and Z. Geng, “Statistical approaches for causal inference (in Chinese),” *Sci Sin Math*, vol. 48, pp. 1753–1778, 2018.

[51] M. Waldmann, “The oxford handbook of causal reasoning,” Oxford University Press, Oxford, United Kingdom, 2017.

[52] J. Zhang, L. Yu, F. Xingbing, X. Yang, X. Gang, and Z. Rui, “Model of the intrusion detection system based on the integration of spatial-temporal features,” *Computers & Security*, vol. 89, 2020.

[53] M. A. Hawawreh, N. Moustafa, and E. Sitnikova, “Identification of malicious activities in industrial internet of things based on deep learning models,” *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.

[54] M. A. Ferrag, L. Maglaras, S. Moschoyiannis et al., “Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study,” *Journal of Information Security and Applications*, vol. 50, Article ID 102419, 2020.

[55] W. Haider, J. Hu, J. Slay, B. Turnbull, and Y. Xie, “Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling,” *Journal of Network and Computer Applications*, vol. 87, pp. 185–192, 2017.

[56] Z. Chen, Q. Yan, H. Han et al., “Machine Learning based mobile malware detection using highly imbalanced network traffic,” *Information Sciences*, vol. 433, pp. 346–364, 2018.

[57] N. V. Chawla, K. W. Bowyer, L. O. Hall et al., “Smote: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[58] X. Tan, S. Su, Z. Huang et al., “Wireless sensor networks intrusion detection based on smote and the random forest algorithm,” *Sensors*, vol. 19, no. 1, 2019.

[59] J. Li, K. Cheng, S. Wang et al., “Feature selection: a data perspective,” *ACM Computing Surveys*, vol. 50, no. 6, p. 94, 2018.

[60] M. A. Hall, “Correlation-based feature selection for machine learning,” Ph. D. dissertation, University of Waikato Hamilton, Hamilton, New Zealand, 1999.

[61] A. Moubayed, M. Injadat, A. Shami et al., “Relationship between student engagement and performance in e learning environment using association rules,” in *Proceedings of the 2018 IEEE World Engineering Educati on Conference (EDUNINE)*, pp. 1–6, Buenos Aires, Argentina, March, 2018.