2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) | 979-8-3315-0971-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISPA63168.2024.00278

Robust Malicious Domain Detection Based on Spatio-Temporal Hypergraph Networks

Liangyi Gong*†✉, Kunxian Lv*†, Chun Long*†✉, Hao Fu*†, Huanran Wang‡, Wei Wan*†, Wu Yang‡
*Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China
‡School of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China

***Abstract*—Malicious domains serve as significant resources for adversaries to execute cyber attacks and are crucial indicators for detecting network intrusions. In practical scenarios, malicious domains associated with various attacks are intermingled within DNS traffic, leading to variability in the performance of machine learning-based detection methods. To address this challenge, we have collected extensive DNS traffic data spanning 12 months from a real-world large-scale network with 1 million users. From this dataset, we have extracted numerous requested domains, encompassing 267 attacks that exploit malicious domain names. Furthermore, we have observed that the distinct properties of malicious domains associated with different attacks contribute to the fluctuating performance of machine learning-based detection models. Consequently, we have introduced a spatiotemporal hypergraph network model, which establishes high-order relationships among domain properties to enhance the generalization capability and robustness of the detection model. The results of extensive testing experiments demonstrate that our model achieves remarkable performance, with an average precision of 97% and recall of 98%.**

***Index Terms*—Malicious Domain Detection; Hypergraph Networks; Spatio-Temporal; Correlation Analysis; High-order Association.**

I. INTRODUCTION

The domain name system (DNS) plays a pivotal role in Internet services, achieving the accurate mapping between domain names and Internet resources, while the domain has also become an important resource for adversaries to launch attacks. Various network attacks adeptly create dynamic mappings between domain names and malicious resources to locate control servers flexibly. Therefore, malicious domain detection has become an indispensable function of network intrusion detection [1], which helps security operators find potential network intrusions, monitor the intrusion process, and take interception measures effectively.

In mainstream network intrusion detection systems, malicious domain names are usually configured as a block list as a detection rule. However, due to the inherent flexibility of domain name mapping mechanisms, attackers can bypass detection using DGA [2] or Flux technology. Therefore, many scholars have begun to explore the use of machine learning techniques to detect mutated malicious domain names [3]. Existing research works select internal and external features of domain names and analyze the mixture or correlation of these features to identify malicious domain names [4]. Nevertheless, practical security operations show that machine learning-based malicious domain name detection has issues of unstable performance and significant fluctuations in detection accuracy. Therefore, this instability poses a challenge to the widespread adoption of machine learning-based malicious domain name detection technology in daily large-scale network security operations.

To comprehend the reasons for the performance decline of machine learning models in real-world applications, we collect extensive malicious domain query records for measurements and analysis. Specifically, we gathered DNS traffic from a large-scale network with 1 million users, extracted the requested domains, and identified the malicious domains by a high-quality threat intelligence business system. Over 12 months, we captured about 52 billion domain query records and identified more than 6.4 million malicious domains involving 267 types of attacks. Moreover, we filter out benign domains based on the rankings of several popular websites. After data deduplication, a total of 1 million benign domains and about 0.25 million malicious domains are collected as our datasets used in the research. Finally, we use network management commands (NSlookup and WHOIS) to query the resolution records and registration information of both benign and malicious domains.

Then, we evaluated the detection performance of 12 different popular machine-learning algorithms [5] on the datasets. The results reveal significant variations in the detection accuracy of different machine-learning algorithms across diverse attacks. We postulate that this phenomenon is primarily attributed to the vast variation in the characteristics of malicious domains associated with diverse attacks, thereby posing challenges in accurately embedding and fusing these features within the Euclidean space. Consequently, traditional machine learning algorithms encounter difficulties in simultaneously identifying various types of attack malicious domains during large-scale assaults [6], resulting in a diminished level of robustness. Hence, there is a compelling need to delve into the development of a novel malicious domain detection model that relies on high-order correlation analysis.

Therefore, we introduce a spatiotemporal hypergraph

2158-9208/24/$31.00 ©2024 IEEE
DOI 10.1109/ISPA63168.2024.00278

network model, building high-order associations based on multi-dimensional features to enhance the generalization and robustness of the detection model. Firstly, the domains composed of characters are encoded by a dictionary-encoding approach. Next, we propose a multi-view hypergraph learning model to achieve the high-order interactive learning of multi-dimensional domain features. To be specific, the encoded domain names in one time block are converted into a matrix, and then extracting high-dimensional semantic features by convolution operations to build hypergraph nodes. Moreover, we find that malicious domain names of similar attacks have complex many-to-many correlation relationships, thus we build hyperedge according to attribute feature to represent the associations among similar domain names. Given that different attribute features of malicious domains play distinct roles in detecting diverse attacks, instead of mixing these heterogeneous attribute features, we adopt a dual attention fusion mechanism to enhance the correlation among similar nodes by combining edge structure and node semantics with different weights.

In practice, we also observe that the three static attributes of some malicious domains are stealthy and easy to forge, so attackers try to exploit the weaknesses of the ML-based detection model to evade the detection. Therefore, we introduce the timing feature to model malicious domain access behaviors. To be more specific, the hypergraph identification results in adjacent time blocks are fused through the temporal networks, so as to introduce the domain access timing feature to our model. The extensive experimental results show that the integration of the timing feature greatly improves the detection accuracy.

The principal contributions of this research are outlined below:

1) Our research focuses on the detection of malicious domains, leveraging large-scale network DNS traffic as a basis. We have conducted a thorough analysis of the factors that contribute to performance degradation when traditional machine learning techniques are applied in extensive, real-world detection scenarios.
2) In pursuit of enhancing the detection capabilities, we have developed a novel spatiotemporal hypergraph network specifically designed for malicious domain detection. This approach utilizes multi-dimensional domain attributes to augment both the accuracy and robustness of the detection process.
3) The implementation of our malicious domain detection model has undergone rigorous testing in online environments over a period of several months. The results demonstrate the stability and exceptional robustness of our detection method.

II. Measurements

A. Data acquisition and annotation

Initially, we obtain the requested domain names from filtering DNS traffic through an intrusion detection system of a large-scale network with 100Gpbs bandwidth and 1 million users. In the measurement, we de-duplicate the domain names and retain about 0.6 million domain names from 6.5 million domain requests per hour to construct the unlabeled dataset *rDNS*. For the purpose of labeling data, we utilize a commercial threat intelligence analysis platform [7] to distinguish the domain names. Ultimately, the *rDNS* contains malicious domains and the types of attacks that correspond to them. The *rDNS* contains 52 billion domain access records for 12 months. Of these, the number of malicious domain names is up to 6.4 million, including 267 types of network attacks. To enrich the *rDNS*, we have requested a blacklist of malicious domains from security vendors that they have not yet made public.

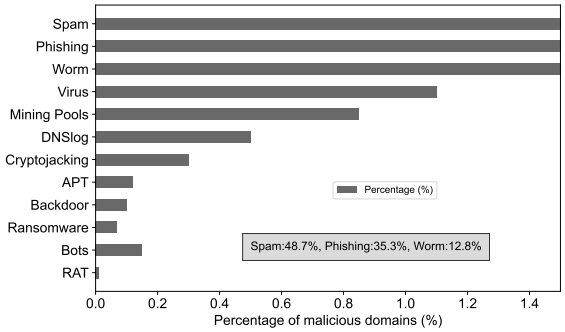

Fig. 1. Attack Percentage in our datasets.

To evaluate the quality of the *rDNS*, we use the following public threat intelligence analysis websites for checking the domain names in the *rDNS*, VirusTool [8], VenusEye [9], 360 Brain [10], Dbappsecurity [11], and Qianxin [12]. We randomly select 10K malicious domain names from the *rDNS* for these websites and the results returned include malicious domain name labels, attack types, and threat levels. The result revealed an accuracy rate exceeding 99% for identifying malicious domains, where a single malicious domain within the test set was identified as such by more than three threat intelligence centers.

We conduct a survey of the mainstream whitelist databases [13]–[15] to find the most suitable for the *rDNS*. As the Transco can provide more comprehensive information of popular primary domain names than others, we conduct the benign domain names in the *rDNS* from Alexa [13] of the mainstream whitelist.

The 267 kinds of attacks in *rDNS* are classified into 12 major categories according to the specific attack mode. The statistical distribution of the *rDNS* is shown in Fig 1. As shown in Fig 1, the largest number of malicious domains is the Spam attack category, accounting for 48.7% of the total number of malicious domains. The least number is

TABLE I
PERFORMANCE EVALUATION OF 14 ML CLASSIFICATION MODELS ON MALICIOUS DOMAIN DETECTION.

Models	Performance			
	Accuracy	Precision	Recall	F1-Score
kNN	0.97	0.95	0.95	0.95
LR	0.96	0.92	0.93	0.93
SVM	0.96	0.95	0.94	0.94
AdaBoot	0.96	0.93	0.92	0.93
GNB	0.87	0.70	0.96	0.80
MNB	0.86	0.95	0.54	0.69
LDA	0.95	0.91	0.92	0.91
QDA	0.94	0.87	0.93	0.90
DT	0.97	0.95	0.97	0.96
GBDT	0.97	0.95	0.96	0.96
RF	**0.98**	**0.96**	**0.97**	**0.96**
CNN	0.95	0.95	0.86	0.91
LSTM	0.84	0.81	0.58	0.67
HAN	0.90	0.89	0.76	0.82

the remote control software category, accounting for only 0.013% of the total number of malicious domains.

B. Machine learning methods evaluation

To understand the limitations imposed by real traffic on malicious domain detection models, we evaluate several popular machine-learning methods on the *rDNS*. Initially, we extract the domain-related features, including lexical features, registration features, and record features. In the paper, we select 27 key features from three types of information, as shown in Table II. To eliminate the impact caused by the structural difference between features, We normalize the features. Specifically, we convert numeric data to $(0, 1)$, and for non-numeric features such as registrar, country, and agency. We use the method of One-Hot encoding, which is 0 if the feature is missing and 1 if it is not.

Moreover, we perform a comprehensive analysis of tree-based models, mathematical models, and common deep-learning models [16]. The evaluation results are shown in Table I. It can be found that tree-based and mathematical models get higher accuracy. As most of the features used for malicious domain detection are discrete features with low relevance, the tree structure methods have a better effect under their ability to classify discrete features well. We think that the fusion and representation of discrete features in Euclidean space is difficult for traditional machine learning methods.

In the experiments, we discovered that the key features used for malicious domain detection vary in diverse attacks. For instance, the hierarchy number of APT attack domain names, which is shown in Fig. 2, is larger than those of other attacks. For registration features shown in Fig. 3, there are obvious differences in the integrity of malicious domain name contacts, registrants, and contact mailboxes in different attacks. Besides, in terms of record features, the NS record entries of cryptojacking attacks are obviously more than those of APT and worm virus malicious domain names, and the A record entries of cyptojacking and Bots attacks are higher than the others, which is illustrated in Fig. 4. In one word, it is found that the malicious domain names of various attack types have obvious differences in the lexical features, register features, and record features.

TABLE II
27 KEY FEATURES USED IN ML-BASED MALICIOUS DOMAIN DETECTION.

Attrubutes	Key Features
Lexical	entropy, length, subdomain length, upper words percentage, numeric percentage, longest numeric length, longest numeric offset, and hierarchy number
Record	No. of A, No. of AAAA, No. of CNAME, No. of NS, No. of MX, No. of primary dns, and refresh rate, retry rate, expiration time, and TTL in the SOA records.
Register	registrar, registration country, registration time, expiration time, registrant email, status, DNSSEC, name server, registry organization

According to the above finding, we select the best one of these methods, the random forest algorithm, to detect malicious domains of diverse attack types. To our surprise, the results show that the algorithm has great differences in malicious domain detection performance of 12 attack types, as shown in Fig. 5. The fundamental challenge that limits the performance of machine learning methods is the ability to express complex features and their relationship in Euclidean space. Therefore, achieving high-dimensional feature fusion and identifying higher-order correlations among domain names become critical to enhance model robustness.

III. MODEL DESGIN & EVAULATION

To overcome the above issue, we propose a HYTAN model, a high-robust malicious domain detection method. The overview of this method is shown in Fig. 6. Firstly, the domain name is encoded to extract the semantic feature by the hypergraph convolution layer. Secondly, we propose a multi-view hypergraph model, building hyperedge according to multi-dimensional features to represent the associations among domains. Meanwhile, we introduce a dual attention mechanism to enhance the similarities among related domains by fusing the features represented by hyperedges and the semantics modeled by hypergraph nodes with different weights. Further, we utilize a temporal module to model malicious domain access behaviors, where identification results in adjacent time blocks are fused through the temporal network. Finally, we use fully connected layer to classify the domain into benign and malicious domains based on the feature generated by the hypergraph sequential convolutional neural network.

A. Domain Encoding

The malicious domains usually disregard memorability and are crafted by scripts that employ numerous meaningless numbers and chaotic characters. Therefore, we utilize construct hypergraph nodes based on the name semantics

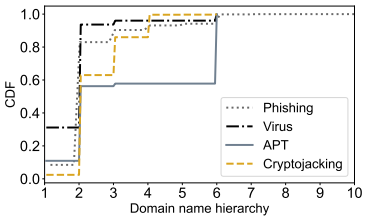

Fig. 2. Hierarchy of domains of different attacks.

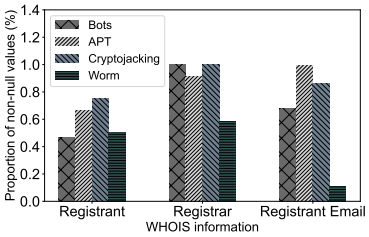

Fig. 3. Registration information of domains of different attacks.

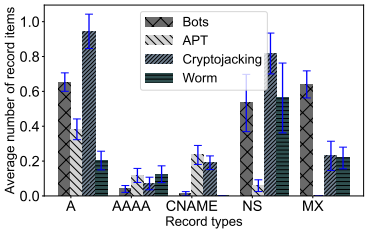

Fig. 4. Record type number of domains of different attacks.

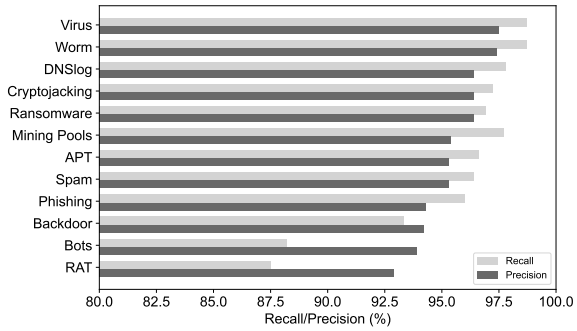

Fig. 5. Performance of RF-based malicious domain detection model on 12 types of attacks.

to deduce the domain generation method and whether the domain is benign or not. In practice, the domains in a time block composed of characters are encoded by a dictionary-encoding approach and converted into a matrix. Then, the multi-scale convolution kernels extract specific local semantic features, followed by the fully connected layers and secondary convolutions to extract global name semantics. Finally, the global features are input into the fully connected layer for dimension reduction, and the results are obtained to form the hypergraph nodes.

B. Multi-View Hypergraph Learning

The relationship among malicious domains is beyond binary, so the traditional graph structure cannot describe the high-order association among domains. As hypergraphs are a generalization of graphs that is suitable for representing complex relationships, the hypergraph is well-suited for modeling higher-order associations among domains. Since each hyperedge can connect any number of nodes, this flexibility can help us describe the complex relationships in the domains more accurately.

Specifically, $G = (V_G, E_G)$ is an unweighted undirected hypergraph, consisting of n nodes and m hyperedges. $V_G = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E_G = \{e_1, e_2, \ldots, e_m\}$ is the set of edges. We use the matrix H to store the hypergraph.

$$H_{j,k} = \begin{cases} 1, v_j \in E_k \\ 0, otherwise \end{cases}$$

Considering that domains of the same attack type have similar attributes, we use the similarity of domain features to build hyperedges. For example, under normal circumstances, attackers register malicious domains in batches, so that the registration information of the domains has the same registrant, similar registration time and the same registration email address. Moreover, large-scale network attacks often come from several organizations. Because of the relatively few control servers, the resolution information of domains may often point to the same IP or CNAME. Therefore, we can build the feature matrix according to the registration features and resolution features of the domain name. In our experiment, a total of 23 features are selected to build the hyperedges. In addition, we regard each feature as a kind of hyperedge division basis, so for the same attribute, domains with the same characteristics are divided into the same hyperedge.

Assume that for a certain attribute A, there are m different attribute values among n domains. Then, according to the different attribute values, the n domains are divided into m hyperedges, denoted as $S_A = \{s_1, \ldots, s_m\}$. The hypergraph incidence matrix of A attributes is denoted as

$$H_{i,j} = \begin{cases} 1, i \in s_j \\ 0, otherwise \end{cases}$$

If the attribute of node i conforms to the value of s_j set, the corresponding value in the hypergraph matrix will be denoted as 1, indicating that the node belongs to that hyperedge.

The domain attributes are either collective or non-collective, this affects our definition of 'same characteristic'. For example, a single domain name resolves multiple IP addresses, this attribute is a collective attribute. While the registrar name attribute must be a single value, this attribute is a non-collective attribute. When there is an intersection of a collective attribute between two domains, we consider that they are two domains with 'the same characteristic', which satisfies the condition of constituting a hyperedge.

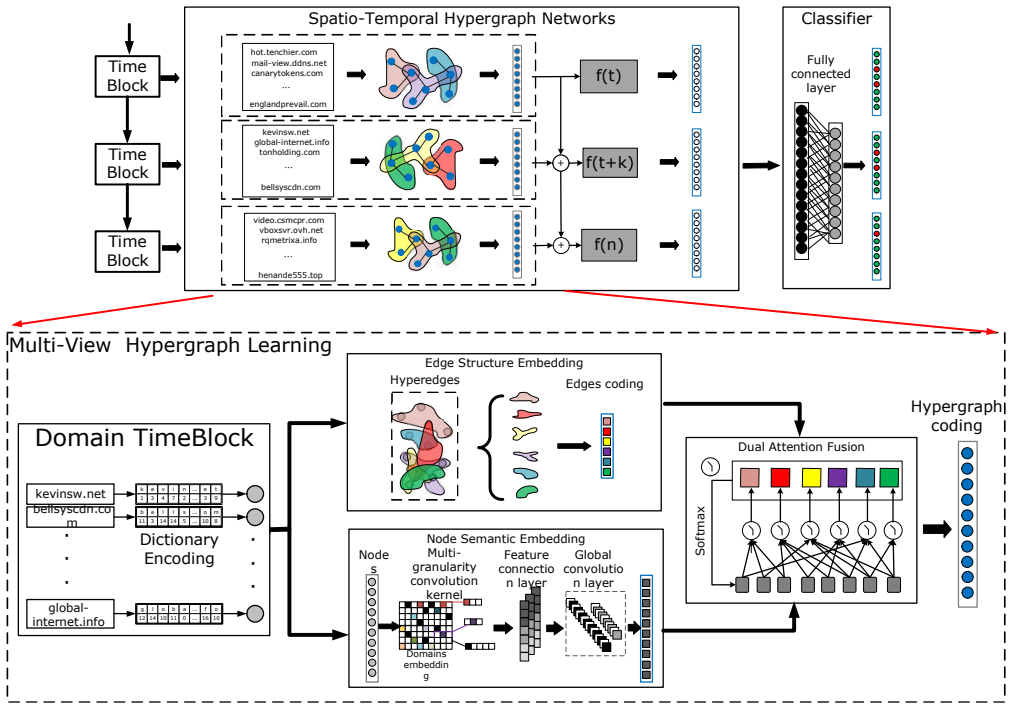

Fig. 6. Architectural overview of of our HYTAN model.

On the contrary, only when two non-collective attributes are exactly equal, they are considered to meet the condition of constituting a hyperedge. Finally, we merge the hyperedges of all the attributes to form a hypergraph.

C. Dual Attention Fusion

The numerous domains can be associated with each other through different attributes, which makes neutral domains suspicious and leads to a high false positive rate. Therefore, we introduce the attention mechanism for hypergraph learning, improving the discrimination ability of hypergraph model detection. As different attribute features of malicious domains play distinct roles in detecting diverse attacks, using the same weight for different malicious domains may lead to misjudgment of benign domains. For example, in botnet attacks, malicious domains often resolve to the same IP address, so the IP address is significant among all the resolution features, and the attention of its corresponding hyperedge should be higher than that of other hyperedges. However, for other types of malicious domains, even if some domains resolve to the same IP address, they cannot be determined to be the same class of domains at once. Because the same IP address could be the host IP address in the botnet but also could be that the cloud server randomly assigned the same IP to another domain name.

To focus on all the categories of malicious domains, we propose a novel dual attention mechanisms to feature the propagation process. We combine edge structure and node semantics with different weights to enhance the similarities among related hypergraph nodes. We employ the feature propagation process from nodes to hyperedges and then from hyperedges to nodes, realizing the feature fusion and correlation in the entire hypergraph, as shown in Fig. 7.

Firstly, we extract the non-zero element pairs in the hypergraph G consisting of the feature incidence matrix H. After that, we concatenate the node features X and a learnable parameter matrix W for attention weight calculation. Next, through the node-level attention module composed of a neural network, we obtain the attention weights, subsequently mapping these weights to a dense matrix Q based on the feature incidence matrix H.

$$Q = Map(Attention(X||W), H)$$

Finally, we successively perform the fusion of attention mechanisms and softmax normalization to obtain the attention matrix A from nodes to hyperedges and use matrix A to summarize the hyperedge features.

$$F_{edges} = A_N \cdot X$$

Similar to the process described above, the features propagate from hyperedges to nodes to obtain the attention matrix based on hyperedges-level attention from nodes to hyperedges. We use two different attention mechanisms to fuse nodes and hyperedges respectively. Finally, the attention

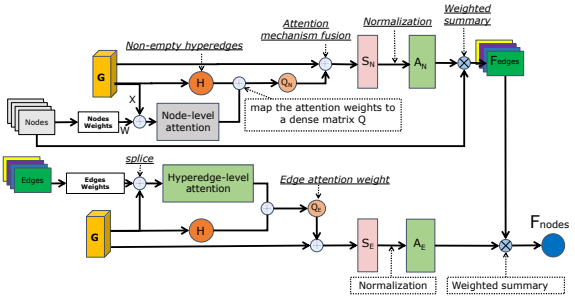

Fig. 7. Attention mechanism of HYTAN system.

matrix is used to sum up the node features to complete a feature propagation process from the hyperedge to the node. The node features are constructed by the above propagation process from their first-order neighbors. Repeating the above hypergraph convolution process, the nodes can finally learn the features in the whole hypergraph.

D. Temporal Convolution Network

As mentioned in Section III, we observed that some malicious domains lack spatial correlation with other domains due to the lack of attributes or rare attack types. As they are represented as isolated points or edge points on the hypergraph structure, so it is difficult to learn their behavior characteristics through the hypergraph network. The reason for isolated points is that attackers intentionally avoid detection and deliberately omit a large number of attributes when registering domains. Since attack domains show periodic or impulsive behavior patterns in time series, we introduce a hypergraph-based temporal neural network learning model to enhance the model's ability to fuse the temporal and spatial features of domains.

First, we construct DNS domain name data in chronological order. $X = \{x_1, x_2, \ldots, x_n\}$ represents the dataset of collected DNS domain name data, where x_i is the domain data at the time i. Then, we partition the dataset into time windows based on the set step size and collect domain name data within sliding time windows. If set the time step as t, then divide the dataset X into a sequence of time windows $\{W_1, W_2, \ldots, W_k\}$, where each window W_i consists of data for M consecutive time points, $W_i = \{x_{(i-1)\cdot t+1}, x_{(i-1)\cdot t+2}, \ldots, x_{(i-1)\cdot t+M}\}$. For the domain data contained in a time window, we perform a complete hypergraph learning process to obtain the hypergraph encoding result. For each domain name dataset $D(M)$ in the time window W_i, we conduct hypergraph learning to generate the hypergraph encoding result H_i. We utilize a temporal neural network to temporally fuse the hypergraph encodings $\{H_1, H_2, \ldots, H_k\}$ from different time windows, resulting in a temporal feature representation F, to ultimately achieve malicious domain classification related to the time series. Finally, we employ the temporal feature F for malicious domain classification related to the time series, i.e., malicious domain classification $R(F)$.

TABLE III
EVALUATION OF DATASETS EXPERIMENTS.

Datasets	Accuracy	Precision	Recall	F1-Score
rDNS 2023	0.98	0.97	0.98	0.98
CIC-Bell-DNS 2021	0.97	0.97	0.97	0.96

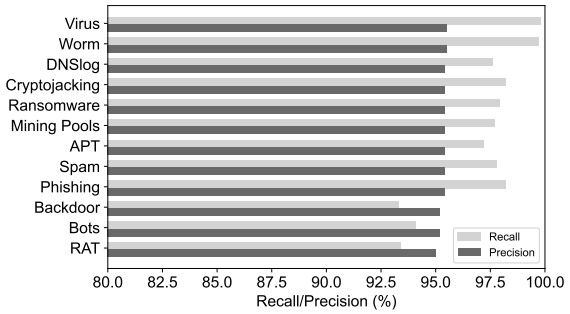

Fig. 8. Performance of HYTAN-based malicious domain detection on 12 types of attacks.

E. Model evaluation

1) Dataset and experiment configuration: To evaluate the performance of the HYTAN model, we test the model on a workstation with Ubuntu 22.04.2 LTS, Intel(R) Xeon(R) Gold 5218 CPU, Quadro RTX 5000 graphics card, and 128 GB RAM, using *rDNS*. The code implementation is based on the neural network framework Pytorch 2.0, and its parameters are set as follows:

1) The convolution kernel size of the character feature embedding layer is 3, 5, and 7, respectively;
2) The output channel number of the concatenated convolutional layer is 128, 256, 1024, and 2048, respectively;
3) GNN $GAT(\cdot)$ outputs 2048 feature dimensions;
4) LSTM (Long short-term memory) is selected as the temporal network of the HYTAN model;
5) The size of $DM(\cdot)$ in the learning layer of the temporal network is set to 512, and the step size t is set to 256;
6) The linear classification layer has two layers, the output dimension of the first layer is 2048, and the output dimension of the second layer is 2.

2) Experimental results: Due to the large number of domains in the dataset, we use 80% of the *rDNS* to train the HYTAN mode and 20% of the data to evaluate its performance. The experimental results show our model can achieve 98% accuracy, 97% precision, 98% recall on average, and the F1-Score is up to 98% on the *rDNS*. Moreover, in order to evaluate the generalization of our model, we also test our model on another large DNS features dataset of 400,000 benign and 13,011 malicious samples. As

TABLE IV
EVALUATION OF MISLABELING EXPERIMENTS.

Model	Noise Ratio	Accuracy	Precision	Recall	F1-Score
HYTAN	0.05	0.96	0.97	0.95	0.96
	0.1	0.92	0.93	0.90	0.92
	0.15	0.84	0.86	0.84	0.85
RF	0.05	0.83	0.82	0.84	0.83
	0.1	0.73	0.76	0.68	0.71
	0.15	0.64	0.67	0.58	0.62

shown in Table III, the HYTAN model still shows significant advantages in accuracy, precision, recall rate and F1 score.

In addition, it can be found that the HYTAN model shows stronger robustness in the detection of malicious domains of diverse attacks in Fig. 5 and Fig. 8. The advantages of the HYTAN model are mainly reflected in the following two aspects. Firstly, the model is based on multi-dimensional character feature extraction and introduces attention and temporal networks, thereby enhancing the spatio-temporal correlation between nodes. These make the model have more obvious discrimination in domain name feature expression and improve the discrimination effect. Secondly, the HYTAN model realizes a high-dimensional fusion of different categories of features based on the multi-view hypergraph model, so it can better distinguish malicious domains of different attacks, improving the robustness of the model.

The machine learning methods are limited by manual feature selection, deep learning methods such as CNN and LSTM require complex network structure design and relational models such as HYTAN lack the ability to express domain name node features in time series. In contrast, the HYTAN model integrates feature fusion, temporal network and attention mechanism to avoid these problems and therefore shows better detection performance and robustness.

3) Performance testing: **Experiments of Vulnerability to Mislabeling:** Due to the concealment and dynamic nature of malicious domains, the training dataset occasionally includes some outdated or inaccurate labels. Such kind of mislabeling often has a great impact on the supervised model. For this case, the vulnerability of the model to mislabeling becomes crucial, which guarantees that the model trained on a dataset including mislabeled data still generates accurate and reliable predictions. In order to test the vulnerability to mislabeling of the model, we set some of the data labels to erroneous labels for training, and finally use the newly trained model for testing, as shown in Table IV.

As shown in Table IV, the detection results of the HYTAN model do not change significantly with the change of labels, but the detection accuracy of the random forest algorithm decreases greatly. This is because the HYTAN model uses the spatio-temporal correlation mechanism for feature fusion. Even if part of the node information is lost, it can still supplement the hypergraph with features provided by other nodes, while the attention mechanism can ignore invalid nodes and edges. In contrast, it is difficult for machine learning algorithms that only perform static feature extraction to process training samples with misleading labels.

TABLE V
EVALUATION OF ABLATION EXPERIMENTS.

Experiment	Accuracy	Precision	Recall	F1-Score
Ablating Hypergraph Networks	0.78	0.77	0.78	0.78
Ablating Attention Mechanism	0.92	0.93	0.91	0.92
Ablating Temporal Networks	0.90	0.91	0.90	0.90

TABLE VI
EVALUATION OF HYTAN WITH DIFFERENT TEMPORAL NETWORKS.

Models	Accuracy	Precision	Recall	F1-Score
LSTM	0.98	0.97	0.98	0.98
GRU	0.97	0.97	0.97	0.97
Transformer	0.97	0.97	0.98	0.97

Experiments of Component Ablation: In order to further explore the influence of each component in the HYTAN model, we construct multiple ablation experiments, as shown in Table V. Firstly, we conduct the ablation experiment of the hypergraph convolutional neural network. In order to evaluate the impact of hypergraph networks on malicious domain detection performance, we use ordinary convolutional neural networks to replace the hypergraph convolutional neural network components. According to the results, it is found that the detection accuracy based on the ordinary convolutional neural network drops by 21%. This is because the ordinary convolutional neural network is a linear network, which makes it difficult to effectively capture the inherent rich correlation of domain features. On the contrary, the hypergraph networks can express high-order association relationships, which in turn achieve a better detection effect.

Secondly, we conduct the ablation experiments of the attention mechanism. The dual attention mechanism can deepen the network's understanding of the domain features content and effectively reduce the impact of weak correlation on detection performance. In the experiment, we eliminate the dual attention mechanism and only implement the detection model based on the hypergraph networks. The results show that the performance of the new model is 6% lower. In fact, there are obvious differences between hyperedges of different attributes and different nodes of various attacks. When the same weight is used for feature propagation, the features of diverse malicious domains in the data cannot be expressed differently, which will further reduce the robustness of the model in practice.

Thirdly, we conduct the ablation experiment of the temporal network. Temporal neural network is an important part of the HYTAN model. By incorporating temporal features, we can boost the correlation of isolated points and the representation of sparse graphs, thereby improving the model's memory capabilities. In the experiment, we reduce the temporal features and only keep the attention

TABLE VII
EVALUATION OF PARAMETER TUNNING EXPERIMENTS.

Batch Size	Step Size	Accuracy	Precsion	Recall	F1-Score	Time (ms)
32	16	0.80	0.83	0.70	0.77	3627
64	32	0.85	0.88	0.80	0.84	3661
128	64	0.93	0.93	0.95	0.94	3848
256	128	0.96	0.97	0.95	0.96	3961
512	**256**	**0.98**	**0.97**	**0.98**	**0.98**	**4032**
1024	512	0.98	0.97	0.98	0.98	9135
512	128	0.95	0.94	0.96	0.95	1283
512	384	0.98	0.98	0.98	0.98	7932

hypergraph network component. The results show that the model detection accuracy is reduced by 7% when the time features are missing. With the time series, the model can learn long-term sequence stream data instead of static data in a certain time, and therefore strengthen the ability to learn historical attack characteristics. In addition, we also evaluate the performance of the HYTAN using different temproal network models. As shown in Table VI, compared with GRU (Gate Recurrent Unit) model, both LSTM and Transformer models can achieve higher detection precision and recall. Because the computation of Transformer model is relatively larger, we finally adopted LSTM model as the temproal network component of HYTAN.

Experiments of Parameter Tuning: The parameters of the HYTAN model mainly contain the size of the batch size and the step size in the temporal networks. Theoretically speaking, if the batch size is too small, it will not be able to form an effective correlation network, so the detection accuracy will decrease. On the contrary, if the batch size is too large, the complexity of the network will increase greatly and therefore the detection efficiency will decrease. In addition, a small step size can enhance the temporal correlation, but it will increase the detection time. Conversely, a large step size will increase the model checking efficiency, but it will lose the detection performance. Therefore, we test different parameters for batch size and step size to conduct experiments, and the results are shown in Table VII.

According to the results, we find that when the batch size is set to 512 and the step size is set to 256, the HYTAN model achieves the best detection performance. When increasing the batch size, the model effect does not increase significantly but the time required for detection suddenly increases. Moreover, the improvement of the HYTAN model is not obvious when the step size is reduced.

IV. CONCLUSION

In this paper, we carry out an innovative experiment to solve the problems of unstable recognition performance, low robustness and weak generalization of ML-based malicious domain detection. Moreover, we utilize a high-quality threat intelligence analysis platform, and match more than 6.4 million malicious domain request records involving more than 260 network attack types within 12 months. Next, we compare the detection accuracy of mainstream ML-based malicious domain detection. Furthermore, we propose a novel spatiotemporal hyergraph networks model to fuse high-order associations among domain features, enhancing the generalization ability and robustness of the detection model. Extensive testing experiment results show that our model can achieve an impressive 97% precision and 98% recall on average.

V. ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China under grant no.2023YFB3105700, NSF of China under grant no.62272440, NSFC-Xinjiang Joint Fund Key Program under grant no.U2003206, NSFC-Regional Joint Fund Key Program under grant no.U22A2036 and Research Team Project supported by Natural Science Foundation of Heilongjiang (Grant no.TD2022F001).

REFERENCES

[1] Y. Shi, G. Chen, and J. Li, "Malicious Domain Name Detection based on Extreme Machine Learning," *Neural Processing Letters*, vol. 48, pp. 1347–1357, 2018.
[2] X. Yun, J. Huang, Y. Wang, T. Zang, Y. Zhou, and Y. Zhang, "Khaos: An Adversarial Neural Network DGA with High Anti-detection Ability," *IEEE transactions on information forensics and security*, vol. 15, pp. 2225–2240, 2019.
[3] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on Malicious Domains Detection through DNS Data Analysis," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.
[4] C. Hamroun, A. Amamou, K. Haddadou, H. Haroun, and G. Pujolle, "A Review on Lexical based Malicious Domain Name Detection Methods," in *Proc. of IEEE 6th Cyber Security in Networking Conference (CSNet)*. IEEE, 2022, pp. 1–7.
[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[6] X. Hu, W. Gao, G. Cheng, R. Li, Y. Zhou, and H. Wu, "Towards Early and Accurate Network Intrusion Detection Using Graph Embedding," *IEEE Transactions on Information Forensics and Security*, 2023.
[7] ThreatBook.com, "X Threat Intelligence Community," https://x.threatbook.cn, 2023.
[8] VirusTotal, "Intelligence," https://www.virustotal.com/gui/, 2023.
[9] V. G. Inc., "Venus Eye," https://www.venuseye.com.cn/, 2023.
[10] L. 360 Digital Security Technology Group Co., "360 Brain," https://ti.360.net/, 2023.
[11] l. Dbappsecurity Co., "Security Star Chart Platform," https://ti.dbappsecurity.com.cn/, 2023.
[12] Q. A. X. T. G. Inc., "ALPHA Threat Intelligence Platform," https://ti.qianxin.com/, 2023.
[13] amazonaws.com, "Alexa Top 1 Million Sites," http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, 2023.
[14] C. Umbrella, "Umbrella Popularity List," https://s3-us-west-1.amazonaws.com/umbrella-static/index.html, 2023.
[15] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "A research-oriented Top Sites Ranking Hardened Against Manipulation-Tranco," 2021.
[16] L. Gong, H. Lin, Z. Li, F. Qian, Y. Li, X. Ma, and Y. Liu, "Systematically landing machine learning onto market-scale mobile malware detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1615–1628, 2020.