

Wei Lu · Yuqing Zhang ·
Weiping Wen · Hanbing Yan ·
Chao Li (Eds.)

Communications in Computer and Information Science

1699

Cyber Security

19th China Annual Conference, CNCERT 2022
Beijing, China, August 16–17, 2022
Revised Selected Papers



 Springer

The Springer logo, which consists of a stylized chess knight (horse) facing left, positioned above the word 'Springer' in a serif font.

OPEN ACCESS

Editorial Board Members

Joaquim Filipe 

Polytechnic Institute of Setúbal, Setúbal, Portugal

Ashish Ghosh

Indian Statistical Institute, Kolkata, India

Raquel Oliveira Prates 

Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil

Lizhu Zhou

Tsinghua University, Beijing, China

More information about this series at <https://link.springer.com/bookseries/7899>

Wei Lu · Yuqing Zhang · Weiping Wen ·
Hanbing Yan · Chao Li (Eds.)

Cyber Security

19th China Annual Conference, CNCERT 2022
Beijing, China, August 16–17, 2022
Revised Selected Papers

Editors

Wei Lu
CNCERT
Beijing, China

Yuqing Zhang
University of Chinese Academy of Sciences
Beijing, China

Weiping Wen
Peking University
Beijing, China

Hanbing Yan
CNCERT
Beijing, China

Chao Li
CNCERT
Beijing, China



ISSN 1865-0929

ISSN 1865-0937 (electronic)

Communications in Computer and Information Science

ISBN 978-981-19-8284-2

ISBN 978-981-19-8285-9 (eBook)

<https://doi.org/10.1007/978-981-19-8285-9>

© The Editor(s) (if applicable) and The Author(s) 2022. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The China Cyber Security Annual Conference is the annual event of the National Computer Network Emergency Response Technical Team/Coordination Center of China (hereinafter referred to as CNCERT/CC). Since 2004, CNCERT/CC has successfully held 18 China Cyber Security Annual Conferences. As an important bridge for technical and service exchange on cyber security affairs among industry, academics, and practitioners, the conference has played an active role in safeguarding cyber security and raising social awareness.

Founded in August 2001, CNCERT/CC is a non-governmental non-profit cyber security technical center and the key coordination team for China's cyber security emergency response community. As the national CERT of China, CNCERT/CC strives to improve the nation's cyber security posture and safeguard the security of critical information infrastructure. CNCERT/CC leads efforts to prevent, detect, alert, coordinate, and handle cyber security threats and incidents, in line with the guiding principle of "proactive prevention, timely detection, prompt response, and maximized recovery".

This year, the China Cyber Security Annual Conference was held online from August 16 to 17, 2022, on the theme of "Jointly Safeguarding Digital Information Infrastructure" as the 19th event in the series. The conference featured one main session and six sub-sessions. The mission was not only to provide a platform for sharing new emerging trends and concerns on cyber security, and discussing countermeasures or approaches to deal with them, but also to find ways to join hands in managing threats and challenges to digital information infrastructure. There were over 5.8 million visits received to our online event. Please refer to the following URL for more information: <http://conf.cert.org.cn>.

We announced our call for papers on our official website, after which 64 submissions were received by the deadline from authors with a wide range of affiliations, including universities, research institutions, telecom operators, companies, financial institutions, and NGOs. After receiving all submissions, we randomly assigned every reviewer with five papers, and every paper was reviewed by three reviewers in a single blind manner. All submissions were assessed based on their credibility of innovation, contribution, reference value, significance of research, language quality, and originality. We adopted a thorough and competitive reviewing and selection process which took place in two rounds. In the first round we invited the reviewers to conduct an initial review. Based on the comments received, 34 papers passed and the authors of these 34 pre-accepted papers made modifications accordingly. In the second round the modified papers were reviewed again. Finally, 17 out of the total 64 submissions stood out and were accepted. The acceptance rate was 26.56%.

The 17 papers contained in this proceedings cover a wide range of cyber-related topics, including network intrusion detection, cloud network, data security, cryptocurrency, vulnerabilities, mobile Internet security, threat intelligence, and webpage tempering detection etc.

We hereby would like to sincerely thank all the authors for their participation, and our thanks also go to the Program Committee for their considerable efforts and dedication in helping us solicit and select the papers of quality and creativity.

Lastly, we humbly hope this proceedings of CNCERT 2022 will shed some light for all readers in their forthcoming research and exploration of their respective fields.

October 2022

Wei Lu
Yuqing Zhang
Weiping Wen
Hanbing Yan
Chao Li

Min Yang	Fudan University, China
Xinhui Han	Peking University, China
Bo Lang	Beihang University, China
Christopher Kruegel	University of California, USA
Yang Zhang	CISPA Helmholtz Center for Information Security, Germany
Guoai Xu	Beijing University of Posts and Telecommunications, China
Stevens Le Blond	Max Planck Institute for Software Systems, Germany
Siri Bromander	University of Oslo, Norway
Chao Zhang	Tsinghua University, China
Zoubin Ghahramani	University of Cambridge, UK
Guojun Peng	Wuhan University, China
Dawn Song	University of California, Berkeley, USA
Kangjie Lu	University of Minnesota, USA
Xueying Li	Topsec, China
Yaniv David	Technion, Israel
Senlin Luo	Beijing Institute of Technology, China
Meng Xu	Georgia Institute of Technology, USA
Wenling Wu	Institute of Software, Chinese Academy of Sciences, China
Chao Li	CNCERT/CC, China
Li Ding	CNCERT/CC, China
Huaping Cao	CNCERT/CC, China
Ruiguang Li	CNCERT/CC, China
Yu Zhou	CNCERT/CC, China

Contents

Data Security

An Intelligent Data Flow Security Strategy Model of Cloud-Network Integration	3
<i>Nishui Cai, Zhuxiang Deng, and Hao Wang</i>	
Applying a Random Forest Approach to Imbalanced Dataset on Network Monitoring Analysis	28
<i>Qian Chen, Xing Zhang, Ying Wang, Zhijia Zhai, and Fen Yang</i>	
Brief Analysis for Network Security Issues in Mega-Projects Approved for Data Clusters	38
<i>Shizhan Lan and Jing Huang</i>	
Considerations on Evaluation of Practical Cloud Data Protection	51
<i>Rui Mei, Han-Bing Yan, Yongqiang He, Qinqin Wang, Shengqiang Zhu, and Weiping Wen</i>	

Anomaly Detection

A Self-supervised Adversarial Learning Approach for Network Intrusion Detection System	73
<i>Lirui Deng, Youjian Zhao, and Heng Bao</i>	
Anomaly Detection of E-commerce Econnoisseur Based on User Behavior	86
<i>Yangyu Long, Wei Zhao, Jilong Yang, Jincheng Deng, and Fangming Liu</i>	
CMPD: Context-Based Malicious Parameter Detection for APIs	99
<i>Zhangjie Zhao, Lin Zhang, Xing Zhang, Ying Wang, and Yi Qin</i>	
Webpage Tampering Detection Method Based on BiGRU-CRF-RCNN	113
<i>Xiangyu Fan, Jilong Yang, Wei Zhao, Jincheng Deng, and Fangming Liu</i>	

Cryptocurrency

Detecting Bitcoin Nodes by the Cyberspace Search Engines	129
<i>Ruiguang Li, Jiawei Zhu, Jiaqi Gao, Fudong Wu, Dawei Xu, and Liehuang Zhu</i>	

Research on Bitcoin Anti-anonymity Technology Based on Behavior
Vectors Mapping and Aligning Model 139
Shenwen Lin, Hongliang Mao, Zhen Wu, and Jinglin Yang

Information Security

Improving Deepfake Video Detection with Comprehensive
Self-consistency Learning 151
Heng Bao, Lirui Deng, Jiazhi Guan, Liang Zhang, and Xunxun Chen

Research on Information Security Asset Value Assessment Methodology 162
Xueqin Yang, Peng Yang, and Honggang Lin

Vulnerabilities

Knowledge Graph Construction Research From Multi-source Vulnerability
Intelligence 177
Lin Du and Chuanqi Xu

Mobile Internet

Research and Practice of TCP Protocol Optimization in Mobile Internet 187
Shizhan Lan, Lifang Ma, and Jing Huang

Traffic Analysis

Research on Adversarial Patch Attack Defense Method for Traffic Sign
Detection 199
Yanjing Zhang, Jianming Cui, and Ming Liu

Threat Intelligence

Research on Named Entity Recognition Method of Network Threat
Intelligence 213
Keke Zhang, Xu Chen, Yongjun Jing, Shuyang Wang, and Lijun Tang

Text Recognition


Research on the Recognition of Internet Buzzword Features Based
on Transformer 227
Dawei Xu, Yijie She, Zhonghua Tan, Ruiguang Li, and Jian Zhao

Author Index 239

Data Security



An Intelligent Data Flow Security Strategy Model of Cloud-Network Integration

Nishui Cai^(✉) , Zhuxiang Deng, and Hao Wang

Telecom Park, China Telecom Research Institute, Shanghai 201315, China
cainishui@chinatelecom.cn

Abstract. Cloud-network integration business data flow security is mainly reflected in the business deployment stage and online service stage. First, this paper analyzes the trend of the digital platform technology of the cloud-network integration business system, puts forward an intelligent data flow security strategy model of cloud-network integration, including expert rule judgment system of simple cloud scene and AI algorithm application model of complex cloud scene. Then, this paper studies hierarchical linkage cloud-network integration security operation system based on the security policy model of intelligent data flow and risk monitoring capability system for personal privacy data protection by scenario system based on the security policy model of intelligent data flow. Finally, this paper points out that cloud-network integration intelligent data flow security strategy based on AI algorithms needs to be further studied.

Keywords: Cloud-network integration · Digital operation platform · Hierarchical linkage · Security operation · Data classification · Intelligent data flow · Security strategy · AI algorithm

1 Introduction

The so-called “cloud-network integration” means that the cloud is cloud computing, the network is the communication network, the network is the foundation, the cloud is the core, the network moves with the cloud, and the cloud-network is integrated. Cloud-network integration is China’s digital economy development strategy and enterprise digital transformation strategy with Chinese characteristics [1]. Among them, cloud-network integration is the foundation, cloud-network security is the support, digital platform is the hub, and scientific and technological innovation is the core.

At this stage, the main problems faced by cloud-network operation support means are that the cloud-network operation support system is too scattered, the BMO data of cloud-network operation is not fully connected, the improvement of data enabled cloud-network operation efficiency is not obvious, and the application of AI injection into intelligent cloud-network operation is not widely used [2]. The common goal is to establish an AI enabled digital platform, fully understand the needs of customers, implement data-based decisions, provide digital business service capability and efficient response operation system quickly, and adapt to the rapid development of industrial digitization.

© The Author(s) 2022

W. Lu et al. (Eds.): CNCERT 2022, CCIS 1699, pp. 3–27, 2022.

https://doi.org/10.1007/978-981-19-8285-9_1

The new generation cloud-network operation business system should have key technologies such as digital twinning of cloud-network resources [3], decoupled acquisition and control of atomic power, big data and AI enabling, cloud-network integration security operation. It corresponds to the resource center, acquisition and control center, big data and AI center and cloud-network security operation center of the system.

- Digital twinning of cloud-network resources [4]. The resource center is responsible for digitizing cloud-network operation elements, depicting “cloud, network, edge and end” resource information, realizing unified integrated management of resources and network operation data, and providing standardized end-to-end cloud-network related resource data service capabilities and business service capabilities.
- Atomic power decoupling acquisition and control. The acquisition and control center is an important foundation for cloud-network integrated operation[5].
- Big data and AI empowerment. The big data and AI center is responsible for making full use of big data and AI capabilities, connecting BMO domains and enabling scenario applications, such as security policy model of “intelligent data flow” [6].
- Cloud-network integration security operation. The cloud-network security center is responsible for establishing a hierarchical, domain and hierarchical cloud-network security protection system and a hierarchical and linked cloud-network integration security operation strategy to meet the security needs of data flow in the business deployment phase and establishing the risk monitoring capability of user personal information protection by scenarios to meet the security needs of user personal information and other important data flow in the online service phase.

2 Intelligent Data Flow Security Strategy Model of Cloud-Network Integration

The security policy model of intelligent data flow, as shown in Fig. 1, can call different data flow intelligent models according to different scenario applications, such as the security protection inter layer linkage strategy and control rules of “network moves with cloud and cloud moves with data”, and automatically divide new specific security area boundaries and security levels according to the security linkage between different layers.

a) Intelligent data flow in simple cloud scenes

- Data to flow: data capacity, data classification, protection requirements, etc.
- Data flow analysis of simple cloud scenario: including reasoning and judgment based on boundary constraints and expert rules, data flow strategy and multi scheme decision-making selection, cloud-network characteristic capacity, protection level, unit energy consumption of equipment, etc.
- Application scenario: when the data flow changes in a single cloud or two clouds, the rule-based intelligent model is preferred.

b) Intelligent data flow in complex cloud scenes

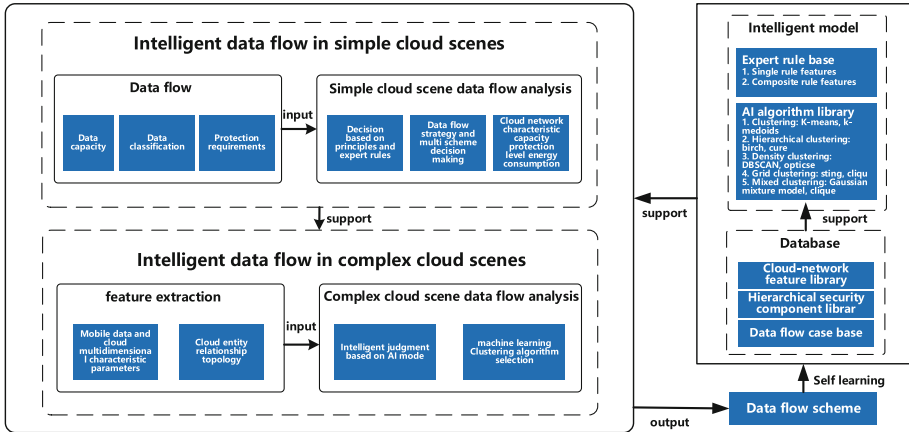


Fig. 1. Security policy model of intelligent data flow.

- Feature extraction: flow data, multi-dimensional feature parameters of cloud and topological relationship of multi cloud entities.
- Data flow analysis of complex cloud scene: intelligent judgment based on AI model and selection of Machine Learning Clustering Algorithm
- Application scenario: when the data flow changes in multiple clouds and there are multidimensional and complex nonlinear characteristics, it is more suitable to apply the intelligent model based on AI algorithm.

c) Database, intelligent model and self-learning

- Database: including cloud feature database, hierarchical security component database and data flow case database.
- Intelligent model: including expert rule base and AI algorithm base. The expert rule base is divided into single feature rule and compound feature rule; AI algorithm, such as:
 - a. Partition clustering: K-means, k-medoids
 - b. Hierarchical clustering: birch, cure
 - c. Cluster density: dbcsi, scan
 - d. Grid clustering: sting, cliqu
 - e. Mixed clustering: Gaussian mixture model, clique
- The self-learning of intelligent model: is to save the output result “data flow scheme” executed by each strategy model to the data flow case base, and then regularly call the latest case base for AI algorithm learning and training, so as to update the relevant model parameters of AI algorithm in time.

3 Hierarchical Linkage Cloud-Network Integration Security Operation System Based on the Security Policy Model of Intelligent Data Flow

Intelligent operation is a new digital operation capability, and it will also be a necessary capability for enterprise digital transformation [7]. At present, intelligent operation needs to gradually realize intelligent operation from single scenario to global intelligent operation.

Aiming at the characteristics of “the network follows the cloud and the cloud follows the data” of the cloud-network integration business system and the security protection requirements of the hierarchical and domain classification of the security domain classification unit of the cloud-network integration business system, based on the research experience of the industry in network and information security strategy, this paper proposes a hierarchical linkage cloud-network integration security operation strategy to meet the security operation requirements of the cloud-network integration business system.

3.1 Cloud-Network Security Protection System with Layers, Regions and Levels

According to the national standard of Chinese information technology GB/T 22239-2019 basic requirements for network security classification protection of information security technology, the security equipment or security components distributed in the network are classified according to “network, cloud, application, data and terminal”, so as to realize the hierarchical decoupling, flexible arrangement and open ability of atomic capability of cloud security resources. The hierarchical security capability components of “network cloud application data terminal” of cloud-network integration business system are shown in Table 1 below.

The security capability components of each layer are as follows:

- a) “Network” layer security capability component.
- b) “Cloud” layer security capability component.
- c) “Application” layer security capability component.
- d) “Data” layer security capability component.
- e) “Terminal” layer security capability component.

3.2 Hierarchical Linkage Cloud-Network Integration Security Operation System Based on the Security Policy Model of Intelligent Data Flow

The hierarchical linkage cloud-network integration security operation system based on the security policy model of intelligent data flow is shown in Fig. 2.

The core modules include: cloud-network integration security policy management point, hierarchical and domain security policy blockchain, “hierarchical linkage” security policy, and hierarchical and domain security control.

- Cloud integrated security policy management point: the security administrator configures the security policy in the cloud integrated security domain unit through the

Table 1. List of hierarchical security capability components of cloud-network system.

Security_capability_component	Security_layer	Safety_level
Terminal virus defense	Terminal	Level 2
Terminal access management	Terminal	Level 3
Terminal data leakage prevention	Terminal	Level 3
Enterprise terminal leakage prevention	Terminal	Level 3+
Data identification	Data	Level 2
Document encryption	Data	Level 2
Data desensitization	Data	Level 3
Database audit	Data	Level 3
Network DLP	Data	Level 3
Data encryption	Data	Level 3
Data destruction	Data	Level 3+
Website content monitoring	Application	Level 2
Web page tamper proof (extranet)	Application	Level 2
Web page tamper proof (intranet)	Application	Level 3+
Mimicry defense	Application	Level 3+
Unified access (4A)	Application	Level 3
Mobile app shell	Application	Level 3
Code audit	Application	Level 3
Anti DDoS	Cloud	Level 2
IPS (internet outlet)	Cloud	Level 2
IPS (intranet outlet)	Cloud	Level 3
WAF (internet outlet)	Cloud	Level 2
WAF (intranet outlet)	Cloud	Level 3
VPN	Cloud	Level 2
Firewall (FW)	Cloud	Level 2
Anti-virus gateway	Cloud	Level 3
Fortress machine	Cloud	Level 2
Vulnerability scanning	Cloud	Level 2
Host protection	Cloud	Level 3+
Honeypot system (extranet)	Cloud	Level 3
Honeypot system (intranet)	Cloud	Level 3+
Full flow (extranet)	Cloud	Level 3

(continued)

Table 1. (continued)

Security_capability_component	Security_layer	Safety_level
Full flow (intranet)	Cloud	Level 3+
Mobile malicious program	Network	Level 2
Network abnormal traffic monitoring	Network	Level 2
Flow direction monitoring	Network	Level 2
Online log retention	Network	Level 2
Stiff wood creep detection	Network	Level 2
Unrecorded website detection	Network	Level 2
Domain name information security management	Network	Level 2
Spam message interception	Network	Level 2
IDC/ISP	Network	Level 2
DNS	Network	Level 3
Attack traceability	Network	Level 3+

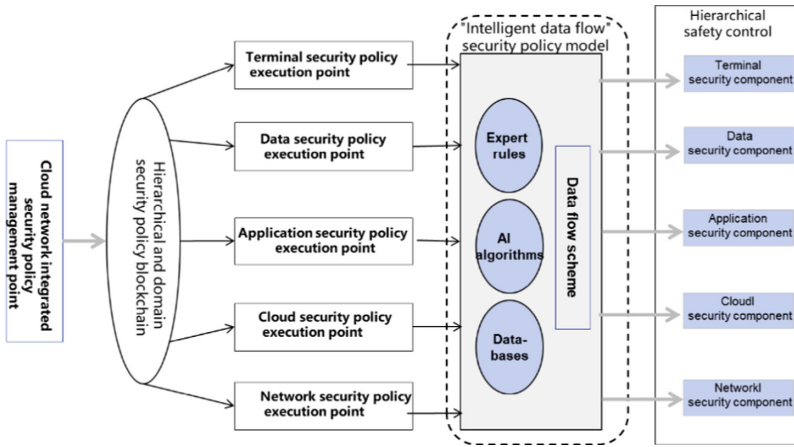


Fig. 2. Cloud-network integration security operation system with hierarchical linkage based on the security policy model of intelligent data flow.

security policy management point, including the setting of security parameters, unified security marks for subjects and objects, authorization of subjects, configuration of trusted authentication policies, etc.

- Hierarchical and domain security policy blockchain: timely release to the security policy execution-point of each layer through the blockchain.
- “Layered linkage” security strategy: the execution-point of each layer’s security strategy is responsible for the query and linkage adjustment of this layer’s security strategy and security control rules; The “layered linkage” security policy rule base can call the

security linkage rules between different layers according to the security protection inter layer linkage policy and control rules of “the network moves with the cloud and the cloud moves with the number”, and automatically divide the new specific security area boundary and security level.

- Hierarchical and sub domain hierarchical security control: implement security policies and security control rules hierarchically, carry out “network cloud application data terminal” hierarchical and sub domain security level protection according to the security level of cloud-network integrated security domain unit, and automatically control the security equipment or security components distributed in the network.

3.3 Feasibility Verification of Cloud-Network Integration “Intelligent Data Flow” Security Strategy

Verification Flow Chart

Hierarchical linkage cloud-network integration security operation flow chart, as shown in Fig. 3.

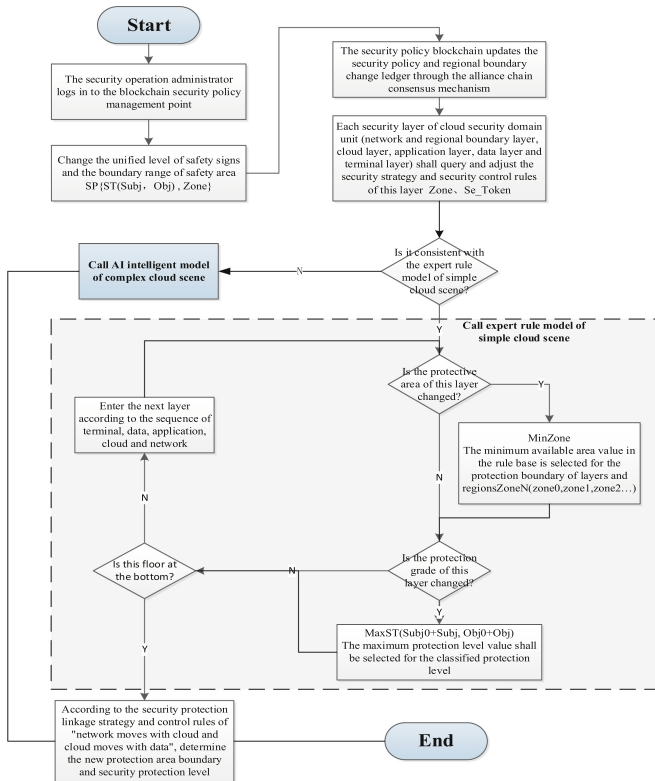


Fig. 3. Hierarchical linkage cloud-network integration security operation flow chart.

According to Fig. 3, the following simulation verifies the security strategy of cloud-network integration “intelligent data flow”. Since there are only two clouds in the application scenario, the simple cloud scenario expert rule model will be called in the simulation process.

Application Case of Hierarchical Rules

Through the security policy management point, the security administrator configures the security policy in the cloud-network integrated security domain unit, including the setting of security parameters, unified security marks (*Se-Token*) for subjects and objects, boundary range of security area (*Zone_defense*), authorization of subjects, configuration of trusted authentication strategy, etc.

$$Security_Police\{Se_Token(Subjects, Objects), Zone_defense\}$$

S0: Cloud-Network Characteristics and Initialization Security Policy Parameters of Layered and Domain

1. S01 Existing Cloud Feature “Layered Linkage” Security Policy Rule Base

$$SPRB(zone0, zone1, zone2 \dots)$$

- Cloud C1, 500 GB of available storage space, corresponding network boundaries N1 and N2, and the initial protection level is level 2.
- Cloud C2, available storage space 2000 GB, corresponding network boundaries N3 and N4, initial protection level 2.

2. S02 Initial Security Policy Parameters

$$SP0\{ST(Subj0, Obj0), Zone0\}$$

- Cloud C1, used storage space 400 gb, remaining available storage space 100 GB, corresponding network boundaries N1 and N2, initial protection level 2.
- Cloud C2, unused, remaining available storage space 2000 g, corresponding network boundaries N3 and N4, initial protection level 2. (See Table 2)

Table 2. Cloud-network integration “intelligent data flow” process table (initial state).

Operation	Cloud	Maximum storage space	Used space	Free space remaining	Network boundary	Safety level
Initial state	C1	500 GB	400 GB	100 GB	N1, N2	Leve 2
Initial state	C2	2000 GB	0 GB	2000 GB	N3, N4	Level 2

S1: Cloud-Network Integration Security Policy Configuration

$$SP\{ST(Obj), Zone\}$$

- Operation 1: add 150 GB data storage and related application deployment, with level 3 security protection.
- Operation 2: reduce 100 GB data storage and related application deployment, with level 3 security protection.

S2: Hierarchical and Domain-Based Security Policy Blockchain

Timely release to the security policy execution-point of each layer through the blockchain.

S3: “Layered Linkage” Security Policy Model

1. *S31: Implementation Points of Security Policies at All Levels*

Receive the security policy issued by the security policy blockchain, call S32 layered linkage security policy rules to calculate the minimum protection area (*MinZone*) and security protection level (*MaxST*), form the adjusted overall requirements of cloud-network security protection, then determine the security policy of this layer, and query the security capability components of relevant security protection levels of this layer according to the cloud-network integration layered protection security capability component system diagram and security protection level, And issue relevant security policy adjustment instructions at all levels.

2. S32: “Layered Linkage” Security Policy Rule Base

$$SPRB(zone0, zone1, zone2 \dots)$$

According to the security protection linkage strategy and control rules of “the network moves with the cloud and the cloud moves with the data”, the security linkage rules between different layers can be called to automatically divide the new specific security area boundary and security level(1).

$$\begin{aligned}
 SP &= SP + SP0 \\
 SP &= \{MaxST(Obj0 + Obj), MinZone\}
 \end{aligned}
 \tag{1}$$

a) Operation 1: Add a Protection Object

Scheme 1: C1 first and then C2, and determine the minimum protection area (*MinZone*) according to the capacity of the protected object:

$$400\text{ GB} + 150\text{ GB} = 550\text{ GB}$$

Determine the safety protection level (*MaxST*) according to the highest level of the protected object:

Adjust the security protection level of C1, C2 and corresponding network boundary to level 3.

- Cloud C1, 500 GB of used storage space, no remaining available storage space, corresponding network boundaries N1 and N2, need to be reinforced with protection level 3.
- Cloud C2, 50 GB used this time, 1950 GB of remaining available storage space, corresponding network boundaries N3 and N4, need to be reinforced with protection level 3.

Similarly, scheme 2, C2 first and then C1... and so on. (See Table 3).

Table 3. Cloud-network integration “intelligent data flow” process table (operation 1)

Operation	Cloud	Maximum storage space	Used space	Free space remaining	Network boundary	Safety level
Initial state	C1	500 GB	400 GB	100 GB	N1, N2	Leve 2
Initial state	C2	2000 GB	0 GB	2000 GB	N3, N4	Level 2
Scheme 1 (Operation 1)	C1	500 GB	500 GB	0 GB	N1, N2	Level 3
Scheme 1 (Operation 1)	C2	2000 GB	50 GB	1950 GB	N3, N4	Level 3
Scheme 2 (Operation 1)	C1	500 GB	400 GB	100 GB	N1, N2	Level 2
Scheme 2 (Operation 1)	C2	2000 GB	150 GB	1850 GB	N3, N4	Level 3

b) Operation 2: Reduce Protected Objects
 Similarly..... (See Table 4).

S4: Hierarchical Security Control

After each security operation policy adjustment operation, immediately receive and execute the security policy adjustment instructions of each layer, query the corresponding security capability components in the hierarchical security capability component list of cloud-network integration business system according to Table 1, and automatically control the security equipment or security components distributed in the cloud-network integration system, that is, the network, cloud, application, data, terminal to load and reinforce the corresponding level of safety protection equipment and application safety components respectively.

By adding and reducing protection objects and protection requirements, the protection strategies of "network, cloud, application, data and terminal" of the cloud system have been adjusted automatically and implemented through the hierarchical security control points.

Table 4. Cloud-network integration “intelligent data flow” process table (operation 2)

Operation	Cloud	Maximum storage space	Used space	Free space remaining	Network boundary	Safety level
Initial state	C1	500 GB	400 GB	100 GB	N1, N2	Leve 2
Initial state	C2	2000 GB	0 GB	2000 GB	N3, N4	Level 2
Scheme 1 (Operation 1)	C1	500 GB	500 GB	0 GB	N1, N2	Level 3
Scheme 1 (Operation 1)	C2	2000 GB	50 GB	1950 GB	N3, N4	Level 3
Scheme 2 (Operation 1)	C1	500 GB	400 GB	100 GB	N1, N2	Level 2
Scheme 2 (Operation 1)	C2	2000 GB	150 GB	1850 GB	N3, N4	Level 3
Scheme 1 (Operation 2)	C1	500 GB	450 GB	50 GB	N1, N2	Level 3
Scheme 1 (Operation 2)	C2	2000 GB	0 GB	2000 GB	N3, N4	Level 2
Scheme 2 (Operation 2)	C1	500 GB	400 GB	100 GB	N1, N2	Level 2
Scheme 2 (Operation 2)	C2	2000 GB	50 GB	1950 GB	N3, N4	Level 3

3.3.1 Application Case of Intelligent Multi-cloud Resource Scheduling

Feature Selection in the Sample Space of Resource Scheduling AI Algorithm in Multi-cloud Scenarios

General principles to be followed:

1. Private cloud resources are scheduled and used preferentially. Only when private cloud resources are insufficient can they be dispatched to the industry cloud or public cloud.
2. Sort according to the billing cost of the industry cloud or public cloud, and give priority to the low-cost industry cloud or public cloud.
3. Evaluate the security capability of the public cloud according to the business or data security level, and calculate it as a resource scheduling parameter measure. Data security capability is an important indicator to evaluate the public cloud.
4. Evaluate according to indicators such as public cloud reliability and resource effectiveness, and calculate them as scheduling parameter measures.

The above principles can be used as a basis for evaluating the importance of feature parameters when AI algorithm selects feature space. In order to better reflect the principles of resource scheduling in a multi-cloud scenario, the operation log of each multi-cloud resource scheduling is generalized. Each scheduling operation is taken as a feature

sequence, and features with strong correlation are selected for vector representation, which is stored in the case database as a case training set.

The characteristic variable name conforming to the above scheduling principle is generalized to $\langle c1_free_space \rangle$, $\langle c2_free_space \rangle$, $\langle c1_safety_level \rangle$, $\langle c2_safety_level \rangle$, $\langle c1_unit\ price \rangle$, $\langle c2_Unit\ price \rangle$, which respectively represents the utilization rate, security level and unit price of economic indicators of the cloud. And $\langle demand_cloud_space \rangle$, $\langle demand_safety_level \rangle$ representing resource scheduling requirements. Due to $\langle c1_safety_level \rangle$, $\langle c2_safety_level \rangle$, $\langle c1_unit\ price \rangle$, $\langle c2_unit\ price \rangle$ the relevant features are relatively stable in resource scheduling. The relevant features are not considered temporarily. Here, only important features are considered to form the sample feature space.

Multi-cloud Resource Scheduling Method Based on KNN Algorithm

The advantage of KNN algorithm is that it can deal with classification problems and regression problems. At the same time, it has strong anti-interference and high accuracy. The low efficiency of the algorithm can be avoided by updating the control sample size, which is more suitable for the operation log size of multi-cloud resource scheduling.

Now only the features $\langle c1_free_space \rangle$, $\langle c2_free_space \rangle$ in the feature space are taken, assuming that the unknown samples are serialized as follows:

$(demand_cloud_space, c1_free_space, c2_free_space) = (100, 350, 200)$ take $k = 3$.

Query the training sample Table 5, calculate the nearest neighbor distance, and determine that the samples with ID6, ID7, and ID8 are k nearest neighbor samples. ID6 and ID7 belong to class 2 and ID8 belong to class 1. Thus, this time, they are classified as class 2 and the corresponding policy_ scheme (50, 50), where cloud1 and cloud2 respectively schedule 50 GB of resource space. (See Table 5).

Table 5. Training sample set

ID	Demand cloud space	Demand safety level	C1 unit price	C1 safety level	C1 free space	C2 unit price	C2 safety level	C2 free space	Police scheme (c1, c2)	Class
1	100	2	10	3	200	5	2	700	(100, 0)	1
2	100	3	10	3	250	5	2	600	(50, 50)	2
3	100	2	10	3	350	5	2	350	(50, 50)	2
4	100	2	10	3	750	5	2	100	(100, 0)	1
5	100	3	10	3	300	5	2	800	(0, 100)	3
6	100	2	10	3	400	5	2	200	(50, 50)	2
7	100	3	10	3	350	5	2	250	(50, 50)	2
8	100	2	10	3	450	5	2	100	(100, 0)	1
9	100	2	10	3	500	5	2	850	(0, 100)	3
10	100	3	10	3	350	5	2	550	(50, 50)	2

4 Risk Monitoring Capability System for Personal Privacy Data Protection by Scenario System Based on the Security Policy Model of Intelligent Data Flow

The effective methods for monitoring the personal information protection risk of the business system are as follows:

- First of all, according to the user's personal information protection compliance requirements, the basic model library of user's personal information protection risk monitoring of the business system is established.
- Then, according to the specific situation of the business function process of the business system, the key node view of personal information protection monitoring of each business process of the business system is established.
- Finally, in combination with personal information protection requirements, according to the basic model library of business system users' personal information protection risk monitoring, corresponding risk detection models are allocated to form a scenario specific business risk identification model for risk identification and analysis.

In this way, it not only solves the problem of visual display of key nodes of user's personal information protection in the business system; It also meets the accurate requirements of the risk monitoring model of each key node, thus improving the accuracy and efficiency of the user's personal information protection risk monitoring.

4.1 List of Basic Risk Models for Rule-Based Personal Privacy Protection

The basic risk models for rule-based personal privacy protection are shown in Table 6.

The basic risk models can be divided into five categories:

- a) account risk model.
- b) exposure risk model.
- c) authority risk model.
- d) transmission risk model.
- e) abnormal behavior risk model.

Table 6. List of risk models for rule-based personal privacy protection.

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
1	1, Account risk identification	Password plaintext disclosure	It is found that illegal personnel may obtain the login information of site users through sniffing, listening, URL interception and other means	Restore HTTP login request Identify account and password fields Account and password plaintext	Request header, request body, response header, response body Account (name, username) Password (PWD, passwd) Plaintext, non plaintext
2	1, Account risk identification	Weak password	Find and identify the login information of the system account and password that may be obtained by illegal personnel through dictionary guessing and Internet	Restore HTTP login request Identification account and password fields Match weak cipher Library	Request header, request body, response header, response body Account (name, username) Password (PWD, passwd) Plaintext, MD5, SHA1 and other weak cipher Libraries
3	1, Account risk identification	Unreasonable login authentication method	It is found that illegal personnel may steal the user's personal information without verification through the get mode	Restore HTTP login request Extract login authentication Judge authentication mode	Request header, request body, response header, response body Account (name, username) Password (PWD, passwd) Submit as get Submit in the form of post, and the cookie contains
4	2, Exposure risk identification	Multiple types of personal information access	Discover and identify pages that can access multiple types of personal information, resulting in increased associativity of personal information	Restore HTTP request content and response content Personal information identification Judge personal information category	Request content and response content Data elements (ID card, mobile number, name, etc.) Data classification

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
5	2, Exposure risk identification	Personal sensitive information is not desensitized	Identify information that should be desensitized but not desensitized to identify sensitive personal information	Restore HTTP request content and response content Personal information identification Desensitization message plaintext	Request content and response content Data elements (ID card, mobile number, name, etc.) Plaintext, non plaintext
6	2, Exposure risk identification	Inconsistent desensitization of personal sensitive information types	It is found that data desensitization is not carried out according to relevant regulations during identification transmission	Restore HTTP request content and response content built-in standard personal sensitive information desensitization rules Desensitization rules for non-standard personal sensitive information	Request content and response content Name (Zhang * *, Zhang *, etc., only one digit is displayed) ID number (513425*****4325, *****) Name (Zhang * CAI) ID number (51***1990**4990, etc.)
7	2, Exposure risk identification	File transfer involving personal information	Discover and identify whether personal information is involved in file content during file transmission	Restore http file transfer request file personal information identification	File format (PDF, pptx, xlsx, xls, docx, Doc, RTF, XT, GZ, 7z, RAR) Number of decompression layers (customized) Data element rules

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
8	2, Exposure risk identification	Mass file transfer	Discover that a large amount of unrecognized personal information is transmitted by means of encryption and compression	Protocol resolution restore access log statistics traffic size of file transfer Traffic file threshold	Request header, request body, response header, response body Source IP Content size (custom m)
9	2, Exposure risk identification	Mail transfer involving personal information	Discover and identify that internal personnel may transmit personal information through e-mail	Restore the message transfer log Identify personal information in the message body or attachment Count the number of personal information data	Sender, recipient, title, body and attachment can be downloaded The system has built-in identification rules and supports new addition Source IP
10	2, Exposure risk identification	Too many pieces of data returned at a time	There may be too many pieces of data returned, which may cause the operator to obtain personal information unrelated to the business	Restore the batch access behavior log Extract the number of personal information for this visit Match set threshold	Request content and response content Data element rules Number of entries (customized)
11	2, Exposure risk identification	Single return content size is too large	Operators' single access to data that exceeds business requirements leads to a large number of personal information disclosure	The return content of the URL access request Count the size of returned content per access Match set threshold	Request content and response content Source IP M (custom)

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
12	2, Exposure risk identification	Batch access to files involving personal information	Discover and identify files containing personal information for batch transfer	Restore http file transfer requests Extract personal information from file transfer Threshold value of the number of pieces of personal information contained in the file	File format (PDF, pptx, xlsx, xls, docx, Doc, RTF, XT, GZ, 7z, RAR) Number of decompression layers (customized) Number (customized)
13	2, Exposure risk identification	Personal sensitive information leak desensitization	The URL carries the account parameters. The operator can access the user's personal information beyond his authority by changing the account	Restore the HTTP request content and response content Personal information identification Matched leak desensitization	Request content and response content Data element rules "Desensitized (yes, no) Not desensitized (yes, no)
14	3, Authority risk identification	Horizontal ultra vires	The SQL statement can be traversed and executed, and the operator can access the user's personal information beyond his authority by changing the account	The URL carries the account parameters Replay HTTP request	Account (name, username) Replay (success, failure)

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
15	3, Authority risk identification	SQL statement traversable execution	The SQL statement can be traversed and executed, and the operator can access the user's personal information beyond his authority by changing the account	Log restore Resend SQL executable	HTTP (request content, response content) Data element rules Replay (success, failure)
16	3, Authority risk identification	Interface not recorded	There may be the act of privately developing interfaces to obtain personal information, resulting in the disclosure of personal information	Automatically discover interface assets Matching interface filing list	IP model Interface filing list
17	3, Authority risk identification	Interface unauthorized discovery	The interface for obtaining personal information may be invoked without authorization, resulting in personal information disclosure	Automatically discover interfaces Record interconnection access relationship Matching interface authorization information	IP model Whether the source IP is the host IP Interface authorization list
18	3, Authority risk identification	Expired but not offline interface access	Expired interfaces are secretly called by illegal personnel to obtain personal information, which leads to personal information disclosure	Automatically discover interface interconnection access relationships Record the last access time of the interface Matching interface offline time	Whether the source IP is the host IP time point Last access time of export interface

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
19	3, Authority risk identification	Invalid account access	When an employee transfers or leaves the company, he / she will immediately withdraw all his / her job numbers, access rights and other permissions, and delete relevant passwords, which may lead to illegal personnel using invalid account numbers to obtain personal information, avoid inspection, and lead to personal information disclosure	Account withdrawal Last access time of account Matching account life cycle	Account (name, username) Password (PWD, passwd) time point Export account list
20	3, Authority risk identification	Account is not authorized to access personal information	Strictly control the high-risk permissions. Do not query the user's personal sensitive information without the user's authorization. Unauthorized access to personal information by the account may cause personal information disclosure	Account withdrawal Count th Count the URL fingerprints accessed by this account URL fingerprints accessed by this account Count the types of personal information accessed by this account Matching account authorization information	Account (name, username) Password (PWD, passwd) URL list Data identification rules Export sensitive account list

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
21	4, Transmission risk identification	IP accessing private network from public network	External personnel invade the intranet system through vulnerabilities, which may lead to personal information leakage or intranet security risks	Identify the server IP Client IP ownership Judge IP ownership	IP home IP home Client IP (public network) Host IP (intranet)
22	5, Abnormal behavior risk identification	Account remote login	The data interface is limited by IP address authentication and illegal login times. The account may be leaked or overstepped, resulting in personal information leakage and increasing the difficulty of tracing	Account withdrawal Client IP address home Illegal zone login	Account (name, username) Password (PWD, passwd) IP address home list Account IP address pool
23	5, Abnormal behavior risk identification	IP multiplexing	The data interface is limited by IP address authentication and illegal login times. The account may be leaked or overstepped, resulting in personal information	Account withdrawal Number of login accounts of the same IP Set threshold	Account (name, username) Password (PWD, passwd) Time range (custom minutes) Number (customized)
24	5, Abnormal behavior risk identification	Account reuse	There are multiple accounts operating on the same IP, which may have been leaked or used horizontally beyond their authority	Account withdrawal Number of login IPS per account Set threshold	Account (name, username) Password (PWD, passwd) Time range (custom minutes) Number (customized)

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
25	5, Abnormal behavior risk identification	Account database deletion	The use of accounts by multiple people may lead to uncontrollable key operational risks of the business system, resulting in cross permissions, increasing the difficulty of accountability and other risks	Restore database operation command Identify database delete operation instructions	Database protocol restore Delete, drop, etc
26	5, Abnormal behavior risk identification	Key instruction operation of account database	A role authority administrator should be set to uniformly manage the addition, deletion and modification of role authority, and identify the possible tampering, overwriting, deletion and addition of key personal information in the database, resulting in incomplete or lost personal information	Restore database operation command Identify key database operation instructions	Database protocol restore create \ update., etc
27	5, Abnormal behavior risk identification	IP short-time access frequency exception	Identify the access of the same IP in a short period of time. High frequency access to personal information may cause mass disclosure of personal information	Number of visits Match set threshold	Client IP Time (custom minutes) URL (number of visits)

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
28	5, Abnormal behavior risk identification	Account short-time access frequency is abnormal	Identify the abnormal behavior of the account to access sensitive personal data for many times in a short time, and the high-frequency access of personal information may lead to the mass disclosure of personal information	Account withdrawal Number of visits Match set threshold	Account (name, username) Password (PWD, passwd) Client IP Time (custom minutes) URL (number of visits)
29	5, Abnormal behavior risk identification	IP machine crawler behavior	Identify that a large amount of personal information is obtained through machine behavior and abnormal time nodes. Personal information is crawled in batches by crawlers, resulting in personal information leakage	Cluster analysis Weight proportion Request characteristics	Proportion of get requests and post requests Exception time Abnormal frequency Abnormal access traffic size Reference is null, user agent is not standard
30	5, Abnormal behavior risk identification	Too many IP (short time / single day) access data	We should focus on auditing the batch operation of key data, and identify the behaviors of illegal access to too much personal information within a limited time	Number of accesses Set threshold	Time (custom minutes) current day Number of entries (customized)
31	5, Abnormal behavior risk identification	IP (short time / single day) access data size is abnormal	The batch operation of key data shall be audited to identify the abnormal behavior that the amount of access data exceeds the daily access value in a short time	Traffic size Set threshold	Time (custom minutes) and current day Number of entries (customized)

(continued)

Table 6. (continued)

ID	Risk classification	Model name	Model function description	Model usage	Include parameters
32	5, Abnormal behavior risk identification	Abnormal number of IP (short time / single day) downloaded files	Identify the abnormal behavior of downloading a large amount of personal information within a limited time of an IP, which is easy to cause a large amount of personal data leakage	Access time period Set threshold	Time (custom minutes) and current day Number (customized)
33	5, Abnormal behavior risk identification	Abnormal number of IP data acquired during non working hours	Identify and find that there are abnormal times of accessing services in the network during non working hours, accessing a large amount of data, and abnormal behaviors of data leakage	Access time period Set threshold	Division of non working hours Number of entries (customized)

4.2 Risk Identification of Personal Privacy Data Protection in Complex Scenarios

Risk Identification of Personal Privacy Data Protection in Complex Scenarios Based on Rules

Batch information export is an important and complex scenario for personal privacy data protection. Here, it is simply divided into two stages: authentication and authorization and information export. It is shown in Fig. 4.

Batch Information Export Scenario Risk Monitoring Process

Step 1: establish batch information according to the management requirements and export the scene management requirements feature matrix.

Step 2: data identification and analysis, that is, access monitoring business system scenarios, mirror business system scenarios, user access traffic data, batch export related multi log multi-dimensional data modeling, including approval, bank mode, permission range and other data for feature rule modeling.

Step 3: risk identification based on rule model, extract and identify models according to key data requirements through protocol analysis and request data analysis, including

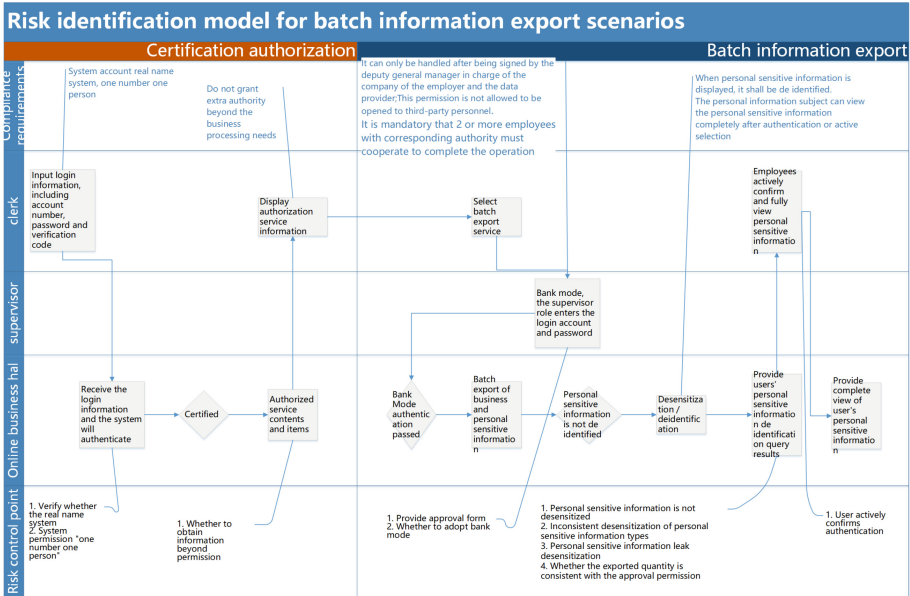


Fig. 4. Batch information export.

data type, regular expression, eigenvalue matching, rule matching, behavior matching, etc.

Step 4: risk identification of AI model by scenario, comparative analysis of various scenario features based on AI and big data technology, and identification of risk scenarios - batch export of AI model feature matching algorithm. Through UEBA user behavior analysis technology, according to the behavior baseline of big data statistical analysis, judge whether it belongs to abnormal behavior derived from batch information, and identify corresponding risks.

Step 5: analyze the authentication model, approve the score scenario information, and judge the compliance of scenario behavior - compare and identify the access behavior and risk of batch exported user information.

Step 6: optimize AI algorithm model to realize self-learning. Based on AI technologies such as machine learning and NLP, the AI algorithm model, strategy and feature base are derived by iteratively optimizing batch information.

5 Conclusion

The security domain unit of cloud-network integration business system has the characteristics of “network cloud application data terminal” layered and sub-domain hierarchical protection and “network moves with cloud and cloud moves with data”. This paper puts forward the intelligent data flow security strategy model of cloud-network integration, including expert rule judgment system of simple cloud scene and AI algorithm application model of complex cloud scene, which can be applied to hierarchical linkage cloud-network integration security operation system and risk monitoring capability

system for personal privacy data protection by scenario system. With the acceleration of enterprise digital transformation and the massive growth of cloud-network integration services, AI algorithm application model of complex cloud scene is an important content of in-depth research in the field of intelligent security operation of cloud-network integration in the next stage.

References

1. Guiqing,L.: Build a new generation of cloud-network operation system and accelerate cloud-network integration. In: Proceedings of the 5th Future Network Development Conference, Nanjing, Jiangsu (2021)
2. Tao, W.: Intelligent connection and cloud-network integration. In: 2021 Huawei Day0 Forum, MWCS, Shanghai, pp. 04–07 (2021)
3. Bing, Q.: The Way of Intelligent Operation and Maintenance - Application Practice Based on AI Technology, 1st edn. China Machine Press, Beijing (2022)
4. Hong, Y.: In the digital age, security begins with attack surface management. In: 2022 Network Security Operation Technology Summit, Beijing (2022)
5. Yanli, G.: How Does the Digital Twin City Promote the New Smart City, 1st edn. China Academy of Information and Communication, Beijing (1999)
6. LNCS Homepage. <http://www.springer.com/lncs>. Accessed 24 May 2022
7. Yalin, Y., Lina, Y., Shuai, R., Qian, Z.: Research on network security protection strategy. In: Proceedings of the 2nd International Conference on Robots and Intelligent System (ICRIS), Warsaw, Poland, pp. 152–154 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Applying a Random Forest Approach to Imbalanced Dataset on Network Monitoring Analysis

Qian Chen^{1,2(✉)}, Xing Zhang^{1,2}, Ying Wang^{1,2}, Zhijia Zhai^{1,2}, and Fen Yang^{1,2}

¹ China Electronics Cyberspace Great Wall Limited Company, Beijing 102209, China
cbxhch@163.com

² National Engineering Laboratory for Big Data Collaborative Security Technology,
Beijing 102209, China

Abstract. Since the rapid growth of big data technology and the continuous development of information technology in recent years, the significance of network security monitoring is increasing consistently. As one of the major tools to secure the system environment, organizations use various monitoring devices to govern the utilities of networks, hardware and applications. Meanwhile, massive and redundant data are produced by these devices constantly, which make a huge problem for analysts and scientists who are willing to extract useful information from them, and even impact the accuracy and efficiency of the monitoring systems. In this paper, we employ random forest algorithm and propose an ensemble learning model under certain scenarios with fixed data features. We use a preprocessing method to balance positive and negative samples, and then use 6 different intrusion detection systems as weak classifiers, which satisfy the rules of “partial sampling” and “partial features selection” of ensemble learning. Finally, we test three combination strategies, including relative majority voting, weighted voting and stacking, to combine the predictions. Experiments show that stacking has a better performance than the other two, with a score of 98.25% in recall, and achieves a 47.91% precision.

Keywords: Random Forest · Network Security · Monitoring and Analysis · Ensemble Learning · Imbalanced Classification

1 Introduction

In recent years, network security monitoring has developed rapidly and played a significant role in network security. Network security monitoring is the prerequisite of a protected and functional network system. In the context of big data, network monitoring data are produced and altered endlessly. Network monitoring systems not only need to recognize the traditional risks such as spiders, port scanning, webshell, injection attack, advanced persistent threat, and phishing mail, but also have to discover the emerging risks such as privacy disclosure, information leakage, data theft, etc. In order to solve these

problems, it is necessary to integrate the strengths of multiple security systems and platforms, which include internet probe, situation awareness system, internet management system, terminal detection system, database protection system, and so forth. However, these systems and platforms are mostly self-contained, which have fuzzy boundaries and duplicated functions. If their advantages can be combined and weaknesses can be complemented in one mechanism, it will reduce unneeded human labor and increase the overall efficiency.

Ensemble learning is the ideal method for solving the problem. Each intrusion detection system can be treated as a weak classifier to distinguish normal and intrusive data, through the integration of several weak classifiers, it will generate a strong classifier with more precise results and higher effectiveness.

In 2001, Giacinto et al. started to solve intrusion detection problems using ensemble learning method [1]. In 2008, Giacinto et al. proposed an ensemble learning method which could detect and discover unknown types of intrusion [2]. Random forest algorithm has been widely used and approved to be effective in intrusion detection ensembles. The common process is to extract the syntax features from PHP code through text analysis, and then build the webshell detection model [3, 4]. Because webshell contains both behavioral features and static text features, it is possible to build a stronger feature combination by merging behavioral features with text static features [5, 6]. Another method is combining random forest with deep learning to build a network intrusion detection model through deep random forest, which can handle more complex and huge datasets [7].

Researchers in intrusion detection often choose public datasets, such as NSL-KDD, ISC2012, ADFA13, DARPA98, or public repositories such as Github. Most of the public datasets are cleaned and balanced, with a proper balance rate of normal and intrusive data, which is suitable for algorithm research. But these datasets are outdated and are not able to reflect the newest trend in intrusion detection. According to certain scenarios, it is necessary to collect specific data and construct a specialized dataset [8].

In this paper, we use a dataset from recall sampling after desensitization of real data, which is deeply imbalanced. In order to adjust the ratio of different samples in an imbalanced dataset, the primary machine learning solutions are undersampling and oversampling. By adjusting model quality metrics for different categories, we can mitigate model failure caused by data imbalance [9, 10].

2 Network Security Monitoring and Random Forest

2.1 Network Security Monitoring

Network security monitoring is a technology that through collecting and analyzing attack alarms to enhance the responses to network intrusions. To conduct the network traffic analysis, people generally export network flow replica via a private network switch and execute the analytical procedures in a dedicated server. By using data presentation tools, data transmission tools, and data collection tools to analyze network traffic, flow information such as sessions, transactions, statistics, metadata, and alert data can be extracted. By analyzing various types of monitoring data, digital threats and intruders can be controlled to ensure network security.

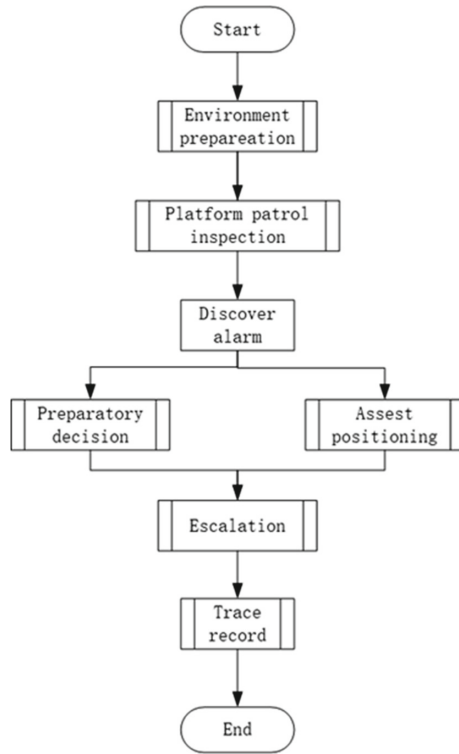


Fig. 1. Network security monitoring process diagram

Figure 1 shows the process diagram of network security monitoring. Network monitoring analysis usually relies on analytical skills of the monitoring staff. Monitoring staff are in charge with extracting information from thousands of alarm data, analyzing and defining the misreporting rate, threaten level and hazard level of each alarm, and implement relevant responses appropriately. In addition to their analytical skill level, monitoring staff also need a thorough understanding of the network environment in specific field, including but not limited to business data patterns, asset locations, etc. They must identify and response to the intrusions timely from numerous alarm data in the complicate environments, and keep tracking the subsequent events and potential risks.

Time is the most important factor in safeguarding the network system. In one sense, misreporting can lead to serious failures because the monitoring staff are unable to deal with intrusions timely. On the other hand, underreporting can cause more risky situations which are hard to predict. Therefore, with the development of network security in recent years, monitoring systems become more and more comprehensive. With the arrival of the big data era and the improvement of computing power, the quantity and repeatability of security data are increased tremendously. New security risks, especially data security risks arise. These factors are challenges to the real-time monitoring.

Because each intrusion detection system has its own technical advantages, the combination of these systems are fairly complex. In practice, administrators must patrol all

intrusion detection systems at the same time during a monitoring process. In addition to intrusion detection tools, monitoring staff should be able to operate other systems flexibly, including asset mapping system, log audit system, host scanning system, security disposal tools, security filing platform, external intelligence platform, tracking and recording platform, and so on. The complex environment and complicated functions also challenge the real-time monitoring.

From the perspective of alarm data itself, in the practical monitoring exercises, most of the alarm information belong to misreported data. Among the alarm data, most intrusions are crawlers, port scanning or vulnerability detection, which are highly repeated and lower threatened. The true threatening invasion are difficult to discover at first because they are hidden in a lot of worthless data.

To face these challenges, a common solution is building network security policies. But policies are always static and fixed, which can be slow to adapt to the network environment changes and cannot be simply applied to all the services and systems.

Based on above conditions, we propose a new solution to decrease the amount of data size and increase the efficiency of security system devices by further screening and classifying of the alarm data.

2.2 Random Forest

Machine learning (ML) has made great achievements in automated classification tasks in recent years, and one of the popular field in ML is ensemble learning. By training multiple weak classifiers and combine them into a strong classifier, ensemble learning can solve a classification problem jointly. Generally speaking, the classifier generated by ensemble learning is more precise than any of the weak classifiers. Sampling methods such as boosting and bagging are commonly used in ensemble learning. As combination strategies, except voting methods such as average method and relative majority voting method, stacking method is also used which integrating and combining models by constructing learners. Random forest is an important method widely used in ensemble learning.

Random forest is an integrated classifier based on bagging expansion, and consists of many decision trees. The predictive output of the classifier is combined after each decision tree is classified. Based on bagging, random feature selection is introduced into random forest. In another words, we need to make a random selection for a feature subset before classification, and then conduct the classification task on the subset.

From the perspective of machine learning, we can treat intrusion detection as a classification task. Intrusion detection systems can transform raw data into structured data tables using data representation tool, then classify data according to attack features and attack types after analyzing them. Each intrusion detection system can be treated as a weak classifier to execute classification. Because of the inaccuracy of the classification result of each weak classifier, we can generate an integrated classifiers using ensemble learning to improve the precision.

By further studies on the data features of intrusion detection in monitoring analysis, we found that each intrusion detection system can only identify part of the attack features because different intrusion detection systems come from different manufacturers with different application scenarios. On the other hand, the network traffic capture method

can be consider as a special bagging in machine learning because each intrusion detection system is deployed at different positions of the network system, which captures incomprehensive and overlapping network traffic data. Therefore, when we use individual intrusion detection system as a classifier, it naturally satisfies the two properties of “partial sampling” and “partial feature selection”. Based on the ideology of random forest, we use ensemble learning method to conduct network monitoring analysis from multiple intrusion detection systems.

Compare with other analysis types in network security, monitoring analysis has higher requirements on timeliness. Analysis with large-scaled neural networks require expensive equipments to ensure the efficiency of computing process. Some algorithms such as K-NN and SVM are only applicable to analysis with small-scaled datasets. With random forest, the computation cost is at equivalent level as the cost of IDS. When considering timeliness, cost and efficiency, and datasets scale, random forest method is the best choice in practice of large-scale network monitoring analysis.

2.3 Imbalanced Learning and Cost-Sensitive Learning

From the perspective of machine learning, the characteristics of monitoring data are typical category imbalance and cost sensitive data. In the network traffic, the vast majority of traffic comes from normal network services, only a small part comes from intrusions.

In this paper, alarm data is regarded as positive class, normal traffic is regarded as negative class. Without any data processing, after sampling the network traffic, we found that the ratio of alarm data to service data reached a level of $1:10^6$ at most. There is a serious imbalance between positive and negative data, which will lead to the natural bias of classification algorithm towards negative data.

Monitoring data is an important data related to network security. The consequences of incorrect classification of monitoring data are different, misreporting may not lead to direct consequences, but underreporting may lead to security vulnerabilities in actual monitoring. As shown in the Table 1, for the confusion matrix, the impact of underreporting is far greater than that of misreporting.

Table 1. Classification result confusion matrix

Real category	Predictions	
	1	0
1	True Positive (TP)	False Negative (FN)
0	False Positive (FP)	True Negative (TN)

There are data level methods and algorithm level methods to solve the class imbalance. The data level methods mainly include oversampling, undersampling and composite sampling. Among them, the disadvantage of undersampling is that it may cause the loss of information, while the disadvantage of oversampling is that it causes over fitting. The algorithm level method is mainly to modify the existing algorithm to pay more attention to the minority class.

In this paper, for the monitoring datasets, we mainly use the undersampling method. Specifically, we use two types of methods. First, we limit the recall channel and increase the proportion of positive samples as we keep the sampling comprehensiveness as much as possible. Second, based on the monitoring data itself, we screen data by several methods include filter white list data, remove data containing important business features, and remove normal network traffic in combination with external intelligence base. After undersampling, positive and negative classes form a data ratio within 1:100.

In order to balance the cost of underreporting, we adjusted the weight of underreporting in the learning process and increased the punishment.

3 Application of Random Forest Algorithm

3.1 Experiment Description

In this paper, the number of service data (TN) is far more than that of other classes, to avoid the disturbance of service data, we use precision, recall and F-score to evaluate classifier performance. Precision is defined as $\frac{TP}{TP+FP}$, recall is defined as $\frac{TP}{TP+FN}$.

From Table 1, FN stands for the number of underreporting, FP stands for the number of misreporting. Considering the importance of underreporting, we increase the weight of FN and define F-score as $\frac{(a^2+1)PR}{a^2P+R}$, in which $a = 2$.

In this paper, we use three different combination strategies to combine classifiers, including relative majority voting, weighted voting and stacking.

For weak classifier h_1, h_2, \dots, h_6 and collection of category tags $\{c_1, c_2, \dots, c_6\}$, we express the prediction output of h_i out of x as a 6-dimensional vector $(h_i^1(x), h_i^2(x), \dots, h_i^6(x))$, let $h_i^j(x)$ be the output of h_i on category tag c_j .

Relative majority voting:

$$H(x) = c_{\arg_j \max \sum_{i=1}^6 h_i^j(x)} \quad (1)$$

Weighted voting (w_i is the weight of h_i):

$$H(x) = c_{\arg_j \max \sum_{i=1}^6 w_i h_i^j(x)} \quad (2)$$

Stacking: A new dataset is generated from the training results of the initial dataset as a training sample, which is called a secondary training set, then we generate secondary learners for training by cross validation.

3.2 Data Sampling and Preprocessing

We select part of the network traffic through the recall channel for analysis. To ensure data comprehensiveness, we need to sample from the complete time period for forming a dataset. Table 2 shows the basic features of the dataset:

Table 2. Features of sampling data

Number of samples	Number of positive	Number of negative	Features
28,445,724	8,558	28,437,166	30

The dataset in Table 2 is generated and sampled from the full period in proportion based on the above features. In a complete period of one week, we observed that during working hours, the network traffic is large and mainly internal business data, while during night and holiday, the amount of network traffic data is relatively small, and the external network access data is the main data. After the dataset is formed, 30 data features are extracted from it combined with each intrusion detection device.

Before we preprocess the data, the ratio of the number of positive classes to the number of negative classes reaches 1:3322, which would cause bias that the results of the model tend to be negative class and cannot be classified correctly when we directly classify on the dataset of Table 2.

Therefore, we clean the dataset in Table 2 by filter the white list, clear the analyzed data in the security policy, remove the business characteristic data and analyze in combination with the external intelligence base. After above preprocess, Table 3 shows the features of the dataset:

Table 3. Features of preprocessed data characteristics

Number of samples	Number of positive	Number of negative	Features
606,267	8,558	597,709	30

After the above preprocess, the ratio of positive and negative classes in the dataset in Table 2 is reduced to nearly 1:69 in Table 3. The following is a further analysis based on the dataset formed in Table 3.

3.3 Classifier Analysis

Combined with intrusion detection equipment, six weak classifiers are extracted from the dataset. By manually analyzing the real situation of positive classification and manually labeling, the actual performance and classification ability of each weak classifier are obtained. Details are shown in the Table 4:

Further analysis based on the data in Table 4:

Table 4. Features of weak classifiers

Identifier	Number of positive	Features	True positive	Precision
1	3,608	15	2,273	63.00%
2	1,550	11	783	50.51%
3	3,685	19	677	18.37%
4	1,904	17	329	17.27%
5	1,812	16	382	21.08%
6	849	9	176	20.73%

- (1) In Table 4, the first two classifiers correspond to Internet traffic, the last four classifiers mainly correspond to each intranet service and terminal detection equipment. Due to the partial overlap of functions, there is a large amount of duplicate data between different classifiers.
- (2) After removing the duplicate alarm classification, there are 4420 external network alarms, in which there are 2629 are correctly classified, the precision is relatively high, reaching 59.48%. The precision of each intranet service and terminal detection equipment is relatively low, there are 4138 alarms, in which there are 791 are correctly classified, the precision is 19.11%.
- (3) On the whole, belong 8558 alarms, there are 3420 are correctly classified, the precision is 39.96%. Among them, the external IP intrusion classifier has high precision, which can identify most conventional intrusions, the most common intrusion includes port scanning, Weblogic attacks, deserialization attacks, and crawlers. The precision of Intranet service management and terminal alarm are relatively low, most of the false positives come from incorrect SQL injection identification, which is because different databases have different management strategies and release orders.
- (4) Through further analysis of the duplicate data, it is found that for some attack features, special devices are required to classify correctly, which makes it possible for us to correctly schedule the dominant classifiers for detection and recognition by combining strategies.

3.4 Combination Strategy

We randomly divided the sample data into two subsets: a training dataset and a testing dataset. 70% of the total sample is used as training data to determine the optimal model parameters. The remaining 30% dataset is used as testing data to evaluate the predictive precision. In this paper, we use three different combination strategies for model training, including relative majority voting, weighted voting and stacking. Table 5 shows the classification results under different combination strategies.

The recall rate in Table 5 reflects the number of underreporting of the combination strategy. In the dataset of this paper, 1% recall rate represents about 40 underreports. Therefore, from Table 5 we can see that using relative majority voting or weighted voting

Table 5. Classification result on different combination strategies

Combination Strategy	Precision	Recall	F-Score
Relative majority voting	50.65%	91.15%	35.14%
Weighted voting	48.98%	95.03%	34.68%
Stacking	47.91%	98.25%	34.47%

for the classifier would cover part of the feature recognition ability, which has certain destructiveness to the model when the amount of weak classifiers are limited. Compared with the above two methods, stacking method has higher performance in classification recall rate. Based on specific scenarios, when data features are relatively fixed, stacking can find the correct classification when weak classifiers conflicted with each other.

4 Conclusion

- (1) In this paper, the safety monitoring dataset is generated by desensitizing the data from practical application and sampling in full cycle. After undersampling, we realize the relative balance of sample data. After data cleaning, the data proportion of the sample is reduced from 1:3322 to 1:69 without damage the features of dataset.
- (2) In practical security production, different intrusion detection devices have different feature recognition capabilities. After preprocessing, the overall classification precision was 39.96%. External IP intrusion is easier to be identified, and the classification precision is 59.48%, which can be correctly identified by most detection devices. For intranet service management and terminal security detection, the detection ability of the classifier is low, only 19.11%. In practical application, these kind of attacks are more necessary to rely on the corresponding equipment with the feature recognition ability to analyze and identify specific features.
- (3) Comparing with voting method, stacking has better performance to combine weak classifiers. After stacking, the precision of classification has increased to 47.91%. Limited by the feature recognition ability of the testing equipment, the precision of the model is limited in this dataset. It is necessary to introduce a new detection and recognition algorithm to greatly improve the precision of the model. Whether the classification precision can be further improved needs further research.

The data application detection in this paper is mainly used for off-line analysis. For the real-time detection of monitoring and analysis, how to conduct real-time analysis through the stream processing engine, and how the detection efficiency and effect are, still pending further study and improvement.

References

1. Giacinto, G., Roli, F.: Design of effective neural network ensembles for image classification purposes. *Image Vis. Comput.* **19**(9/10), 699–707 (2001)

2. Giacinto, G., Perdisci, R., Rio, M.D., et al.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Inf. Fusion* **9**(1), 69–82 (2008)
3. Pan, Z., Chen, Y., Chen, Y., et al.: Webshell detection based on executable data characteristics of PHP code. *Wirel. Commun. Mob. Comput.* **2021**, 5533963 (2021)
4. Fang, Y., Qiu, Y., Liu, L., et al.: Detecting Webshell based on random forest with fasttext, pp. 52–56 (2018)
5. Farnaaz, N., Jabbar, M.A.: Random forest modeling for network intrusion detection system. *Procedia Comput. Sci.* **89**, 213–217 (2016)
6. Zhou, A., Luktarhan, N., Ai, Z.: Research on WebShell detection method based on regularized neighborhood component analysis (RNCA). *Symmetry* **13**(7), 1202 (2021)
7. Liu, Z., Su, N., Qin, Y., et al.: A deep random forest model on spark for network intrusion detection. *Mob. Inf. Syst.* **2020**(1), 1–16 (2020)
8. Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *International Conference on Information Systems Security and Privacy* (2018)
9. Lovri, M., Malev, O., Klobuar, G., et al.: Predictive capability of QSAR models based on the CompTox Zebrafish Embryo Assays: an imbalanced classification problem. *Molecules (Basel, Switzerland)* **26**(6), 1617 (2021)
10. Dong, J., Qian, Q.: A density-based random forest for imbalanced data classification. *Future Internet* **14**, 90 (2022)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Brief Analysis for Network Security Issues in Mega-Projects Approved for Data Clusters

Shizhan Lan^{1,2(✉)} and Jing Huang³

¹ School of Software Engineering, South China University of Technology, Guangzhou 510006, China

lanshizhan@gx.chinamobile.com

² China Mobile Guangxi Branch Co., Ltd., Nanning 530012, China

³ EVERSEC (Beijing) Technology Co., Ltd., Beijing 100191, China

Abstract. Network security is an important guarantee for Mega-projects approved for data clusters. It is necessary to comprehensively improve the network security awareness, monitoring, early warning, disposal and evaluation capabilities of Mega-projects approved for data clusters. It makes a comprehensive analysis on the network security issues in Mega-projects approved for data clusters from dimensions of computing facility security, network facility security, combination and scheduling security, network operation service security, data security, network situation awareness, etc. It is set up gradually evolving atomic power security capabilities for building a ubiquitous security network computing brain. It identifies data assets in an active and passive ways, sorts out data assets through in-depth scanning and information completion, supports the formation of preset templates according to AI (artificial intelligence) models, regular matching, keywords, combination rules, etc., classifies and grades data according to data sensitivity, and visually displays them in the form of charts. It forms a multi-layer architecture system that includes the collaborative scheduling of computing networks on the control side, the perception of network convergence on the data side, management and the scheduling of computing resources on the service side, realizes the interaction and supervision of the whole process, all elements and the whole industry chain of computing scheduling, has functions of security perception, monitoring, early warning, disposal and evaluation, and improves the security perception and linkage monitoring capability of cross data center and clusters. Gradually, it builds a coordinated threat handling capability.

Keywords: Mega-projects approved for data clusters · Network security · Network situation awareness · Network security computing brain · Atomic security capability · AI model

1 Introduction

“Mega-projects approved for data clusters” refers to building a new computing power network system integrating data center, cloud computing and big data [1] to orderly guide the computing power demand in the east to the West. According to the demand of

computing power, China promotes the echelon layout and overall development of data centers from east to west; Accelerates the gradual and rapid iteration of “Mega-projects approved for data clusters”. In order to comprehensively boost the development of new data centers, it builds an intelligent computing ecosystem with new data centers as the core, and gives full play to the enabling and driving role of the digital economy, the Ministry of industry and information technology has formulated and issued the three-year action plan for the development of new data centers (2021–2023) [2], and makes every effort to ensure the promotion of the “Mega-projects approved for data clusters” project.

Network security is the premise for the development of “Mega-projects approved for data clusters”. The “Mega-projects approved for data clusters” project urgently needs to improve the ability of network security perception, monitoring, early warning, disposal and evaluation in an all-round way, accelerate the security protection level of data resources in the whole life cycle, improve the ability of computing power security monitoring and scientific scheduling, and cope with the transformation of network attacks from static analysis to dynamic perception, post disposal to prior prevention, single point prevention and control to global joint prevention.

It is oriented to “Mega-projects approved for data clusters” and meets the scenarios of massive data processing and scientific computing; The training reasoning scenario of artificial intelligence model for east digital west training. Promote the successful implementation of the project of “Mega-projects approved for data clusters”, accelerate the transformation of data centers, and provide new momentum for high-quality economic and social development.

2 General Analysis of Computing Power security

For the construction of the security system of the “Mega-projects approved for data clusters” project, it is necessary to refine the security assurance objectives, clarify the access standards for security technical means such as security situation monitoring, traffic protection and threat disposal, deepen policy reform measures and major engineering suggestions in terms of data resource protection and computing resource monitoring and scheduling, and promote the application security of data resource circulation. As the core task of the construction and application of “Mega-projects approved for data clusters”, network security focuses on building a multi-level collaborative supervision platform and monitoring system for basic networks, data centers, data center clusters, cloud platforms and application enterprises, and improving the ability of “Mega-projects approved for data clusters” project to serve economic operation monitoring and industrial digital transformation monitoring.

The architecture of computing power network consists of three levels: computing power infrastructure, arrangement management and operation service. The infrastructure layer consists of computing infrastructure and network infrastructure to form a new computing network integration infrastructure, and build a flexible and agile computing base and a fully connected intelligent network at the cloud edge. The arrangement management layer realizes the unified arrangement and intelligence of the calculation network by building the brain of the calculation network. The operation service layer

creates a new operation service system and business model by using technologies such as computing power trading, multidimensional dimension and computing power grid connection. In this architecture, safety runs through the whole process, and improving safety endogenous capability has become an important development goal. This paper will analyze the relevant network security issues from the above dimensions.

The overall goal is to build a network security value system of “Mega-projects approved for data clusters” and provide refined, ubiquitous and original twin security services; Build a ubiquitous security computing network brain, provide synchronous “pay as you go” security experience, transform application-based into task-based, and realize differentiated security experience of more refined process. Realize near source defense mode based on twin computing power mode, and realize super edge plus near source side defense mode (Fig. 1).

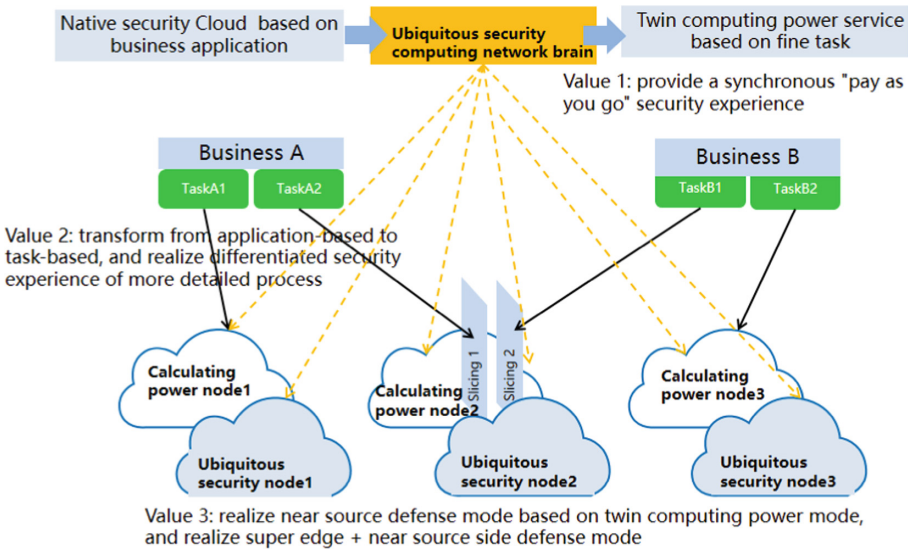


Fig. 1. Ubiquitous security computing network brain

With the rapid development of computing network technology and the continuous integration with Internet+, industrial Internet, big data, cloud computing and other new technologies, more and more information assets provide services with the help of Internet technology. At the micro security capability implementation level, they build a gradually evolving atomic capability security means to protect network security from all dimensions (Table 1).

Table 1. Atomic security capability table

Name of atomic security capability	Atomic security capacity of corresponding resource pool
Host asset discovery	Security asset management system -> host asset discovery
Software asset identification	Security asset management system -> software asset discovery
Web vulnerability scanning	Web vulnerability scanning
Host vulnerability scanning	Vulnerability scanning
Web page tamper proof	Web page tamper proof
Weak password scanning	Terminal detection and response -> weak password scanning
Webshell scan	Terminal detection and response -> webshell scanning
Baseline configuration check	Terminal detection and response -> safety baseline check
Terminal access control	Terminal detection and response -> host network access isolation
Configuration reinforcement (consolidated patch management)	Terminal detection and response -> configuration reinforcement
Document monitoring and protection	Terminal detection and response / Web page tamper proof
Terminal data leakage detection	Terminal detection and response -> terminal data leakage detection
Access behavior audit	Terminal detection and response -> access behavior audit
Terminal intrusion detection protection (combining terminal threat detection and host intrusion detection)	Terminal detection and response -> Terminal intrusion detection / protection
Backup recovery	Terminal detection and response -> backup and recovery
Host Forensics	Terminal detection and response -> host certificate
Terminal antivirus	Terminal detection and corresponding -> anti virus
Network access control (combined network attack suppression)	Next generation Firewall -> access control

(continued)

Table 1. (continued)

Name of atomic security capability	Atomic security capacity of corresponding resource pool
Network address translation (NAT)	Firewall -> network address translation
Network isolation switching	Firewall -> network isolation switch
Network Intrusion Prevention	Next generation firewall
Denial of service protection	Anti denial of service system -> denial of service protection
Sensitive data leakage prevention	Intrusion protection system -> sensitive data protection
Spam protection	Mail Security Gateway -> spam protection
Network virus defense	Network virus defense
Network threat detection	intrusion detection system
Network data leakage detection	Intrusion detection system -> sensitive data outgoing detection
Web application protection	web application firewall
Code audit	Code audit system ->code audit
Database audit	Database audit
Log audit	Log audit
Network security audit	Network security audit
Sensitive data identification	Data security system -> sensitive data identification
Desensitization of sensitive data	Data security system -> sensitive data desensitization
information service	Threat Intelligence Platform -> intelligence service
VPN access	VPN
Operation & Management access control	Fortress -> access control
Name of safe atomic capability	Honeypot -> network attack entrapment

3 Computing Facilities Securities

Computing infrastructure includes cloud computing, edge computing and end computing. While providing powerful computing technology support services for upper tier applications, it also faces many risks. It is necessary to build a comprehensive, systematic and three-dimensional protection means for cloud computing, edge computing and end computing.

In cloud computing, security protection should be provided for physics, virtualization, business, data, operation and maintenance management, etc. In terms of edge computing, security protection should be provided for network services, hardware

environment, virtualization, edge computing platform, applications, capacity opening, management, data, etc.

In terms of end-to-end computing, security protection should be carried out for physics, virtualization, application, capacity opening, management, data, etc.

At the same time, it is also necessary to do a good job in the security protection of cloud, edge and end interconnection, including identity authentication, traffic monitoring and audit, interface control, security situation monitoring and other security protection means.

Simultaneously carry out the basic system planning of computing network security. Based on the independent collaborative evolution stage of the computing power network, strengthen the construction of basic atomic capabilities. Start the standardization of computing security, formulate standardized interfaces and access criteria, and solve the problems of self security and interoperability of computing network. Meet the personalized and distributed computing power needs of customers, conduct technical pre research and pilot demonstration, and adopt decentralized and security identification/security slicing technology to make the security capability compatible with the distribution of computing power; Research on the application of dynamic intelligent network slicing technology to ensure differentiated network service capability.

4 Network Facility Security

SRv6 (segment routing IPv6) simplifies the network protocol type, has good scalability and programmability, can meet the diversified needs of more new services, provides high reliability, and has a good application prospect in cloud services. SRv6 and the new generation SD-WAN (software defined wide area network) are the core technologies to realize the convergence of computing and networking. The networking scheme combining the two can realize the network linkage between the backbone network and enterprise sites, and realize the interconnection and perception of computing power; Deterministic network technology provides quality of service guarantee for new services with ultra-large bandwidth, ultra-low delay and ultra-high reliability. However, the complex network environment, fuzzy security boundary and highly sensitive time delay have also brought new security challenges.

Traditional security solutions do not have the good scalability and programmability of SRv6 and the performance, flexibility or interconnection required for SD-WAN connection. The atomic security capability can support flexibility, interconnection, scalability and programmability, sense the changes of edge connections, and provide consistent policy implementation. This policy can isolate users, applications, workflows, or data based on many parameters to provide security over the entire transaction path. Traffic can be forced to follow specific behaviors, or isolated to specific users or destinations to ensure consistent policy application and execution.

5 Arranging and Scheduling Security

Facing the highly complex computing network environment, the arrangement management layer cooperatively schedules the resources of each domain of the computing

network according to the diversified and customized computing power requirements. The arrangement management layer perceives and cooperates with the arrangement of computing power users, computing tasks, network resources and computing power resources. The arrangement management shall have the ability to control the security of computing power and solve the problem of computing power abuse. The abuse of computing power includes illegal mining, violent cracking and other acts, which not only encroach on computing power resources, but also may use computing power to launch security attacks. Based on the self adaptation mode of the computing power network, establish the North-South linkage between security services and computing power, and promote the scheduling of security computing power. Considering the introduction of heterogeneous computing power nodes rather than completely self built, it is necessary to solve the identity and trust problems of computing power nodes, and conduct research and verification on technologies such as differential privacy and homomorphic encryption during the interaction between algorithms and computing power. Carry out the pre research on node collaboration. The computing power is in multiple nodes. The nodes need to have a synchronization mechanism. The nodes need to adopt an adaptive and self-organizing architecture. The “edge by edge collaboration” mechanism is used for local interaction of capability and performance information.

6 Operation Service Security

Operation service security is mainly to ensure the security of computing network services, including identity security, operation security and integrated application security. Among them, identity security ensures that the identities of computing nodes and users in the computing power network can be identified and verified; The operation security realizes the functions of security transaction, security monitoring, security audit, etc. The integrated application security provides flexible, dynamic and end-to-end business security for differentiated application scenarios such as digital life, intelligent production and digital society.

7 Data Security

Data security [3] runs through all levels of the computing power network, mainly including data asset identification, data security protection, data flow security, computing security, East West training, etc. which can effectively ensure that the data is in an effective and legitimate use state in the whole life cycle.

Data Asset Identification

Data asset identification combines initiative and passivity to discover assets including servers, relational databases, non relational databases, interfaces, etc., and complete the completion of data asset attributes through information completion and in-depth scanning. From the perspective of data assets, data is obtained from SMC/SMP and data resource scanning discovery, and the data is classified and managed at different levels. The classification and classification list management function mainly includes data classification and classification list, important data list and sensitive data list. Real

time display of classification and classification data information of different dimensions, data sorting of identified asset data, classification and classification mapping of data according to data sensitivity, visual display in the form of charts, and controllable storage of warm and cold data.

According to the data classification and grading rules of countries, industries or enterprises, preset templates can be formed according to AI models, regular matching, keywords, combination rules, etc. you can also configure classification and grading templates according to the needs of the current business.

East Digital West Training

The data value evolution path with knowledge as the core. Driven by technology, it develops artificial intelligence model training and reasoning, and constructs the overall technical framework of “East digital West training”. Driven by technology, AI has become the base of new infrastructure technology, promoting the acceleration of artificial intelligence deployment.

AI modeling is different from data development. It has no hierarchical modeling restrictions. At the same time, it opens the way of data reading and warehousing, and supports free modeling; In addition to the basic data processing components, it also has built-in rich machine learning algorithms. It also supports user-defined processing components to help dig deep into data value.

Data Flow Security

Data flow involves data aggregation, data transmission between providers and users, as well as the use of data out of the control of owners. Data will face greater security risks, including personal information disclosure, data vulnerable to attack and disclosure, illegal over collection, analysis and abuse of data, etc. During the data flow process, the data shall be identified, the data flow node, operation, flow direction and other information shall be recorded, and a unified cross domain and cross system data flow identification shall be established to realize that the data flow direction can be controlled and the data flow can be perceived. In order to monitor the flow of data in real time, it is necessary to strengthen network security monitoring through technical means, especially automated security monitoring, and comprehensively monitor and analyze the data sharing platform and system through traffic, logs, configuration files, etc., so as to facilitate early warning and collaborative defense of network security events, and improve the overall security situation awareness, security decision-making and other capabilities.

8 Situational Awareness

Situational awareness integrates detection, early warning, response and disposal functions, and is the safety brain in the active defense system. It plans the security capability of the integrated computing service system of “Mega-projects approved for data clusters”, integrates the existing data center security data, interoperability monitoring platform and supporting business systems, builds a data center level, data center cluster level and industry-wide computing security perception and monitoring platform, realizes the interaction and supervision of the whole process, all elements and the whole industry

chain of computing scheduling, and has the functions of security system [4] perception, monitoring, early warning, disposal, evaluation, etc., Improve the security awareness and linkage monitoring capability of cross data center and cross data center clusters. Gradually build a coordinated threat handling capability.

8.1 Situation Awareness of Network Security Quality Based on Data Network Collaboration

It will improve the monitoring system for computing network governance, promote the optimization of the network architecture and traffic routing of data centers in the eastern and western regions, promote the quality monitoring of data network collaboration, promote the networking of edge data centers, and continuously improve the network capacity of data centers (Fig. 2).

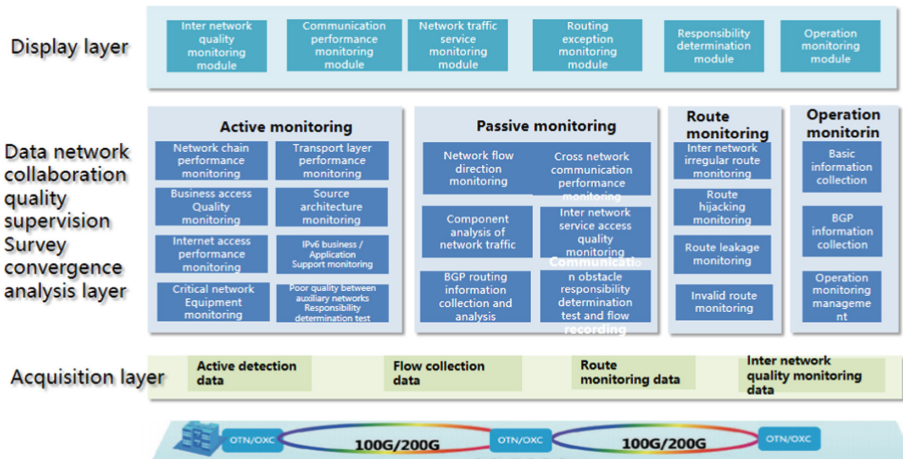


Fig. 2. Situation awareness of network security quality based on data network collaboration

8.2 Situation Awareness of Computing Capacity Security Improvement Evaluation

Take the cloud with the network and use the network to strengthen computing, realize the enhancement of computing power value based on the computing power network, and ensure the enhancement of computing power value with computing power security. Promote the development of computing power network from multiple demands, support the implementation of ubiquitous computing power with multiple technologies, and enhance the security value of computing power with multi-dimensional security situational awareness (Fig. 3).

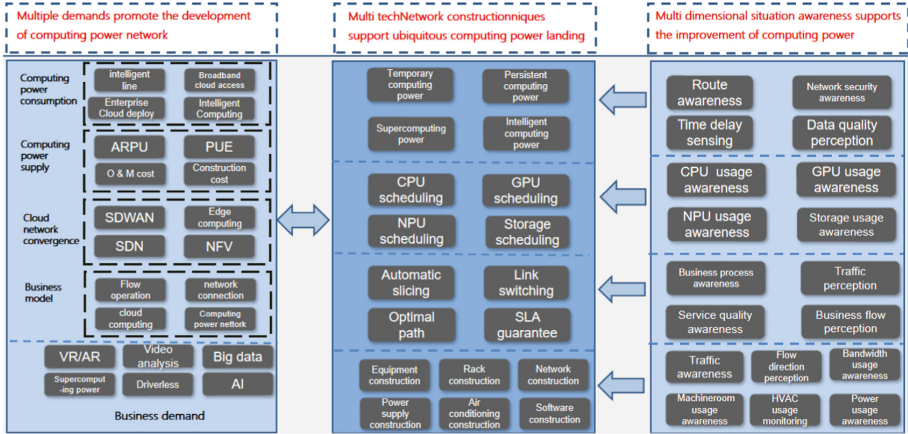


Fig. 3. Situation awareness of computing power security improvement evaluation

8.3 Situation Awareness of Industry Chain Security Enhancement Assessment

Accelerate the key technology and product innovation of the new data center operation security management and other software layers, as well as the cloud native and cloud edge integration security and other platform layers, and improve the software and hardware synergy; Establish and improve the new data center security standard system; Draw the security map of the whole industry chain of the new data center, promote the completion of key links, and carry out the security capability evaluation of the new data center (Fig. 4).

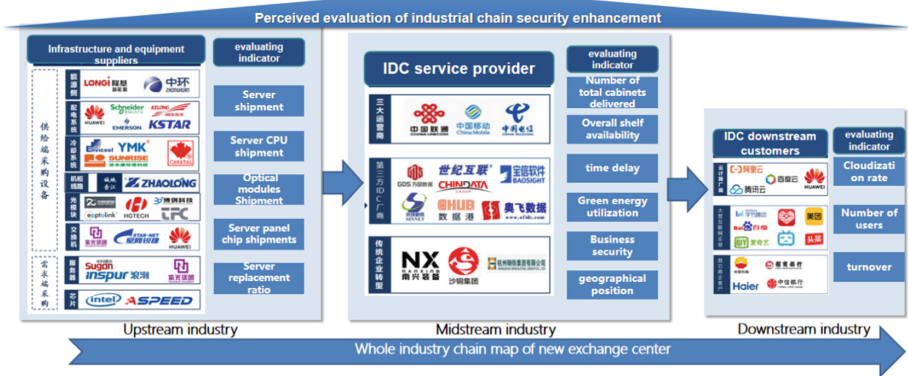


Fig. 4. Industry chain security enhancement assessment situation awareness

8.4 Situation Awareness of Green Low Carbon Assessment

The continuous deepening of the national “double carbon” strategy has put forward higher requirements for the green and low-carbon level of the data center industry, and the PUE, cue and other energy efficiency indicators are more strictly restricted. “Mega-projects approved for data clusters” is a powerful driving scheme for the data center to achieve “carbon neutralization and carbon peak”. The collection and evaluation of energy consumption indicators saved after “Mega-projects approved for data clusters” can be used as one of the dimensions to evaluate the situation awareness of “Mega-projects approved for data clusters”. In order to quickly achieve the “double carbon” goal, implement the notice of the Ministry of industry and information technology on printing and distributing the three-year action plan for the development of new data centers (2021–2023), and optimize the green development of the data center industry chain, it is necessary to establish and improve the green data center standard system (Fig. 5).

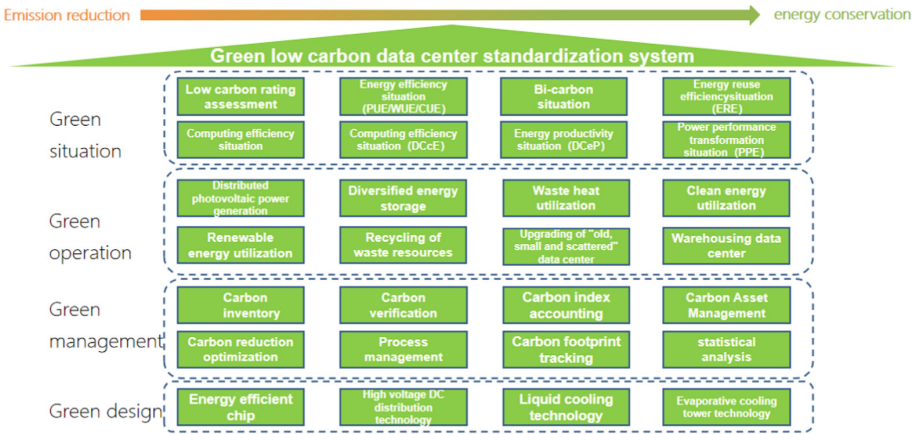


Fig. 5. Green low carbon assessment situational awareness

8.5 Situation Awareness of Security Assurance Assessment

In the important supporting support construction scheme of “Mega-projects approved for data clusters”, it is clearly emphasized that from the aspects of data risk identification and protection, data security compliance assessment, to data encryption protection and related technical monitoring, it is necessary to “synchronously plan, construct and use security technical measures to ensure business stability and data security (Fig. 6).

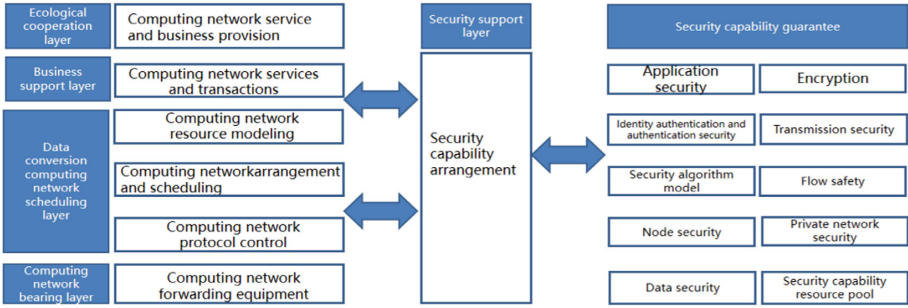


Fig. 6. Security assurance assessment situation awareness

8.6 Threat Collaborative Disposal Scenario

Based on the new data center security monitoring means, facing the "Mega-projects approved for data clusters" network threat collaborative disposal scenario [5], carry out closed-loop disposal and collaborative linkage of threat disposal, deposit the network security risk case base, emergency drill scenario base, emergency disposal plan base, emergency disposal expert base and emergency response tool set base, promote the transformation of threat disposal to risk early warning and pre prevention, and improve the scientificity, accuracy and timeliness of threat disposal, We will strengthen capacity-building for coordinated disposal.

9 Conclusion

The "Mega-projects approved for data clusters" project realizes "the network moves with the cloud, and the cloud moves with the needs", forming a multi-layer architecture system including the computer network collaborative scheduling of the control plane, the network fusion perception and management of the data plane, and the arrangement of computing resources of the service plane.

The new architecture, new technologies and new services of the "Mega-projects approved for data clusters" network may have new security risks that need to be overcome, and need to be guaranteed by a new security mechanism adapted to it. There are potentially complex network risks and computing power node security risks in the infrastructure layer. The scheduling management layer involves scheduling security risks and computing power use out of control. The operation service layer faces problems such as accessing malicious nodes, untrusted transactions, insecure applications, etc. in addition, there may be data security risks such as uncontrollable data flow in the "Mega-projects approved for data clusters" network, which needs to be strengthened through an integrated whole process trusted mechanism.

This paper makes a comprehensive analysis on the network security problems in "Mega-projects approved for data clusters" from the aspects of computing power facility security, network facility security, scheduling security, operation service security, data security, situation awareness and so on. It is proposed to build a network-based ubiquitous

endogenous security system with ubiquitous security computing brain as the core, atomic security capability as the foothold, and intelligent orchestration as the link.

Guided by the security application, oriented to the new business mode of cluster scheduling, combined with the existing traffic protection and security monitoring means in the data center, it focuses on the realization of network security quality situational awareness, computing power security improvement assessment situational awareness, industrial chain security enhancement assessment situational awareness, green low-carbon assessment situational awareness, security assurance assessment situational awareness and other assessment systems for data network collaboration.

And then promote the network convergence, transmission, storage and integration application links for the cluster nodes of the data center to carry out the construction of traffic protection security means; In combination with active detection means and detection work, implement the construction of security situation capability and build the capability of threat collaborative disposal.

References

1. Reply of the national development and Reform Commission and other departments on Approving the Beijing Tianjin Hebei region to start the construction of the national computing node of the national integrated computing network. National Development and Reform Commission document No.(2022)212
2. Three year action plan for new data center development (2021–2023). Ministry of industry and information technology communication document No.(2021)76
3. Kwiecień, A., Maćkowski, M., Sidzina, M.: Data security in microprocessor units. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2013. CCIS, vol. 370, pp. 495–506. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38865-1_50
4. Zhang, Y., Jin, S., Cui, X., Yin, X., Pang, Y.: Network Security Situation Prediction Based on BP and RBF Neural Network. In: Yuan, Y., Wu, X., Lu, Y. (eds.) Trustworthy Computing and Services. ISCTCS 2012. Communications in Computer and Information Science, vol. 320. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35795-4_83
5. Lotz, V.: Threat Scenarios as a Means to Formally Develop Secure Systems. Springer, Berlin Heidelberg (1997)



Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Considerations on Evaluation of Practical Cloud Data Protection

Rui Mei^{1,2} , Han-Bing Yan³ , Yongqiang He⁴, Qinqin Wang^{1,2}, Shengqiang Zhu⁵, and Weiping Wen⁴

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

³ National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), Beijing 100029, China
yhb@cert.org.cn

⁴ Peking University, Beijing 102600, China

⁵ PingAn Cloud, PingAn Insurance Group of China Ltd., Shenzhen 518023, China

Abstract. With the continuous growth of enterprises' digital transformation, business-driven cloud computing has seen tremendous growth. The security community has proposed a large body of technical mechanisms, operational processes, and practical solutions to achieve cloud security. In addition, diverse jurisdictions also present regulatory requirements on data protection to mitigate possible risks, for instance, unauthorized access, data leakage, sensitive information and privacy disclosure. In view of this, several practical standards, frameworks, and best practices in the industry are proposed to evaluate and improve the protection level of cloud data. However, few evaluation models can conduct a comprehensive quantitative evaluation for cloud data protection that includes security, privacy, and even ethical considerations. In this paper, we first make a comprehensive review of cloud data security and privacy issues, especially also including ethical concerns that we consider as a type of specific risks caused by human factors, which refers to acting honorably, honestly, justly, and legally, due diligence, and due care. Then, we propose a novel evaluation model for cloud data protection that can quantitatively assess the protection level. Finally, based on the parallel evaluation between manual assessment by experts and our evaluation model, results show that our evaluation model is consistent with the manual evaluation conclusion.

Keywords: Cloud data protection · Evaluation model · Security · Privacy · Ethics

1 Introduction

With the rapid improvement of cloud computing, the cloud offers flexible and affordable software, platforms, infrastructure, and storage available to organizations across all industries. Faced with limited budgets and increasing growth

© The Author(s) 2022

W. Lu et al. (Eds.): CNCERT 2022, CCIS 1699, pp. 51–69, 2022.

https://doi.org/10.1007/978-981-19-8285-9_4

demands, cloud computing presents an opportunity for organizations to reduce costs, increase flexibility, and improve IT capability [14]. Despite the rapid adoption of cloud computing, security and privacy remain key issues for the security community [20, 25]. Although cloud service providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) continue to expand security services to protect their evolving cloud platforms, security and privacy are ever-lasting considerations while migrating traditional IT to cloud [10].

As we all know, cyberspace is not peaceful, and both external advanced persistent threats (APTs) and insider attacks still occur from time to time. Since cloud environments often contain a variety of tenants and their vast amounts of valuable data, cloud platforms are also targeted by cyber threat actors [32–34]. For external APTs, attackers are always looking for new attack surfaces in the cloud to bypass existing security controls [41]. Insider attacks are often acted by disgruntled insider employees, who have limited authorized access and tend to exfiltrate sensitive data or escalate privilege intentionally. This is also an ethical issue due to the human factor instead of a technical issue.

The implementation of cloud migration by enterprises means losing physical control of systems and data, thus it requires an assessment method to evaluate the protection level of cloud environments, including cloud data. Although many standards, frameworks, and best practices have been proposed by the security community and industry, there is rarely a comprehensive evaluation model that can quantitatively analyze the score of the protection level of cloud data that fully considers security, privacy, and ethical issues. In summary, this paper makes the following contributions:

- We are the first to make a comprehensive review of every aspect of cloud data protection based on our full knowledge of security, privacy, and ethics issues, which consists of technological mechanisms, operational policies, and legal & regulatory compliance.
- We present a novel algorithm to compute the score of protection level based on our insight about important factors that affect cloud data protection, that is, intra-phase, inter-phase, lifecycle operations, and compliance.
- We propose an empirical evaluation model to assess the overall protection level of cloud data based on the score of each factor that affects the protection level.

2 Overview of Cloud Data Protection

This section introduces the methodology related to cloud data protection. We leverage the Data States Model and Cloud Data Lifecycle Model to summarize major security and privacy controls from a top-level perspective. More considerations on fine-grained controls including technique measures, operational policies, and legal & regulatory compliance will be discussed in the rest of the paper.

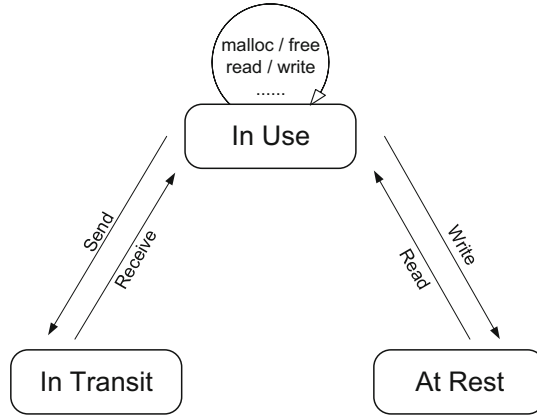


Fig. 1. The data states model in IT systems.

2.1 Data State Model

Data, as a type of critical asset, exists in one of three states both on-premise and in the cloud, including while it is *at rest*, *in transit*, and *in use* [26]. Regardless of the state of the data, IT systems should implement appropriate controls to protect the data and mitigate security and privacy risks [43]. Figure 1 shows the Data State Model, including three states of data and transformation between them. Data in use can be converted to both in-transit state and at rest, however, data at rest cannot be changed to in-transit state directly and vice versa. It is worth noting that this characteristic of conversion between data states depends on the classical Von Neumann architecture which is still the major one all over the world, other computing architectures e.g. quantum computing are out of our scope.

Data in Use, refers to any data in the main memory or other caches while an application is using it. Due to the multitasking and concurrent features of modern information systems, it is important to ensure authorized access to data in use. Operating System (OS) built-in process isolation and application-level sandbox are primary controls for data in memory and cache. However, emerging attack vectors often try to bypass existing security mechanisms by vulnerabilities exploitation or advanced impersonation techniques. To this end, pieces of research attempt to leverage homomorphic encryption [1]. This limits the risk of data leakage because memory doesn't hold unencrypted data.

Data at Rest, aka data on storage, is any data stored on media, such as hard drives, external USB drives, network attached storage (NAS), and storage area network (SAN). The major risks it faces include data exfiltration, integrity breaches, unavailability (e.g. Denial of Service i.e. DoS). Strong symmetric encryption is the key control of data at rest for security and privacy concerns. In Addition, as a compensating control, data redundancy can improve the high availability (HA) of data. Furthermore, strict authentication and authorization controls [30] can also help prevent unauthorized access.

Table 1. Cloud data lifecycle and representative controls

	Six phases of cloud data lifecycle					
	CREATE	STORE	USE	SHARE	ARCHIVE	DESTROY
In Use	×		×	×	×	×
At Rest	×	×	×		×	×
In Transit	×		×	×	×	
Controls	· Data classification · IPSec/TLS VPN	· Encryption · Key mgt.	· Virtualization · DRM/IRM · DLP · IPSec/TLS VPN	· DRM/IRM · IPSec/TLS VPN	· Encryption · Key mgt. · IPSec/TLS VPN	· Crypto-shredding

Data in Transit, also called data in motion, refers to any data transmitted over a network. The exchange of data between information infrastructures almost entirely depends on the transmission network in cyberspace. In particular, unlike the traditional on-premises model using the internal local networks, if enterprises migrate their IT systems to the cloud, all data access will be transferred over the Internet. Therefore, data in transit is more likely to be the target of cyber attacks than the other two data states. Leveraging a combination of symmetric and asymmetric encryption can protect data in transit generally.

2.2 Cloud Data Lifecycle

Data in cloud is constantly being created, stored, used, and transmitted, and once the data is no longer valuable, it needs to be destroyed. Unlike other valuable physical assets, the value of data is time-sensitive and specific, thus data protection is sophisticated. Cloud Data Lifecycle Model provides a generic approach to identifying the broad categories of risks facing the data and associated security or privacy controls, therefore this allows us to consider threats, vulnerabilities, and risks of cloud data at a higher level of abstraction in case of getting bogged down in the concrete details of a specific organization.

Table 1 illustrates each phase of this model and corresponding representative controls. Noting that the cloud data lifecycle is not always iterative, on the contrary, it is not constantly linear, sometimes even exists in multiple phases simultaneously. As an example, data being shared may be used and stored at the same time if co-workers collaborate in a Software as a Service (SaaS) app. Furthermore, data in a phase can also exist in multiple states. Regardless, data should be protected at every stage with security and privacy controls commensurate with its value [18].

- **Create.** Data can be created by a user at on-premises or legacy workstations and then transferred to the cloud, or created directly in the cloud. Data classification [13] is the most important security control at this stage. In addition, if the data is created remotely, transportation security mechanisms such as IPSec/TLS VPN [22] are necessary solutions.
- **Store.** Typically, this phase is synchronized with the creation phase, and encryption is introduced to mitigate threats exposed in the data center of the cloud environment. Meanwhile, for any cryptosystem, key management is also a critical security control.

- **Use.** Unlike data access on-premises, accessing cloud data remotely requires more additional security and privacy controls, including: (1) implementation of virtualization protection controls [44] to ensure that there is no unauthorized access between different guests hosted on the same server; (2) leveraging Digital Rights Management (DRM) aka Information Rights Management (IRM) solutions [40] to implement fine-grained dynamic access control; (3) conducting continuous monitoring via Data Loss Prevention (DLP) solutions [21] which is also used for data classification in *create* phase; and (4) to enable network security transmission mechanisms, such as aforementioned IPSec/TLS VPN.
- **Share.** This phase is relatively self-explanatory, that is, data is granted by its owner to other users or entities that require access. If the data has been highly classified, more accurate and fine-grained access control rules can be provided for data sharing. Conversely, additional controls are required to conduct data protection, many of controls implemented in prior phases will be effective here, such as DRM/IRM solutions, IPSec/TLS VPN, and so forth. Furthermore, due to distributed cloud data centers, data can be located in data centers in different jurisdictions. Thus several restrictions may exist in accordance with regulatory mandates based on the location of the data center.
- **Archive.** As an integral part of the cloud data life cycle, archiving data is used for: (1) Business Continuity/Disaster Recovery (BC/DR) [4, 29]; (2) Data retention and audit [12]; (3) eDiscovery [28]; and other (3) compliance requirements. Similar to the storage phase, the primary security control in this phase is encryption. In addition, due to long timeframes for storage in archives, there may be additional concerns related to availability.
- **Destroy.** Data that is no longer useful and is no longer subject to retention requirements should be securely destroyed. There are many options for data destruction of legacy IT environments or on-premises, e.g. deletion, overwriting, degaussing, etc. However, in the cloud environment, due to data dispersion techniques, there is only one choice to destroy data, which is crypto shredding aka cryptographic erasure [38]. This mechanism refers to encrypting data leveraging one strong encryption engine, then using the key generated by that process, encrypting the data on another different encryption engine, and destroying the key thereafter.

2.3 Shared Responsibility Model

Cloud computing is a business-driven computing model rather than technology-driven, thus the interests of cloud service providers (CSPs) and cloud service customers (CSCs) are not always aligned. CSCs want maximum computing capabilities at the lowest cost. On the other hand, CSPs want to provide as few services as possible while maximizing profits. In this paper, we don't review the cloud computing reference model here, which is clearly defined in the ISO/IEC 17789 [24]. Fortunately, despite the adversarial relationship existing between the two sides, the interests of security and privacy on both sides converge. One example is that a data breach of a CSC caused by vulnerabilities of the infrastructure

Table 2. The shared responsibility model

	IaaS	PaaS	SaaS
Data access security	C	C	C
Application security	C	C	S
Platform security	C	S	P
Virtualized infrastructure security	S	P	P
Physical infrastructure security	P	P	P

in a CSP will bring both parties to suffer brand and reputation damage, lower profits, and even face ongoing lawsuits.

The Cloud Shared Responsibility Model [16] clarifies the clear responsibilities of both the CSPs and CSCs for defense-in-depth of cloud architecture. Table 2 shows details of this model, where the rows and columns represent the layers and cloud service models of the cloud architecture, respectively. In Table 2, cells marking C, S, and P indicate the responsibilities of CSC, Both, and CSP, respectively. It is worth mentioning that although we only list the most common three cloud service models here, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), which are also defined in ISO/IEC 17789 [24], other service models have similar shared responsibility model. We are particularly concerned that regardless of the service model, the responsibility for data access security attributes to the CSC. This means that the ultimate responsibility for any data breach should be borne by the CSC. Of course, the CSC also has the right to seek compensation from the CSP. Due to the elementary principle of “layered defenses” in information security, both CSCs and CSPs need to implement security and privacy controls at different layers to protect data. In a nutshell, this paper doesn’t intend to clearly distinguish which party is responsible for the implemented security controls, which does not influence our evaluation of security, privacy, and ethics of controls.

3 Techniques, Operations, and Compliance

The practice of industry in the past decade shows that a large body of previous excellent approaches, mechanisms, and tools in traditional IT has been introduced for better building the foundation of cloud computing. Therefore, the cloud and traditional IT also share most of the security controls to secure systems and data. These controls usually include three aspects, namely techniques, operational activities, and compliance. We will discuss security and privacy controls for cloud data protection, along with the possible risks and how to mitigate them. Furthermore, ethical considerations are also discussed, which can be also considered as a risk in essence, and refer to acting honorably, honestly, justly, and legally, due diligence, and due care.

The rest of this section will discuss cloud-specific key security and privacy controls involving three aspects i.e. technical mechanisms, operational policies, and legal compliance.

3.1 Technological Mechanisms

Unlike on-premises, the cloud environment can rarely protect data by implementing strong access controls at clear boundaries, thus encryption is the primary option for protecting data. It is known that cloud computing is usually multi-tenant, and even with the deployment model of private cloud, there are also conflicts of interest among different departments within the same organization. Therefore, data obfuscation mechanism is an important security and privacy control. In addition, virtualization techniques and corresponding security controls, as elementary cloud infrastructure, face several cloud-specific risks.

1) *Encryption and Key Management*

It should come as no surprise that cloud computing has a deep dependency on encryption, and no matter what state the data is in, without encryption technology, it is impossible to use cloud computing technology in any secure way. Due to the criticality of encryption organizations should concentrate their efforts on correctly implementing and deploying cryptographic systems, while key management is the area of greatest concern. If an organization uses multiple CSPs or intends to hold physical control over cryptographic keys, one solution is to escrow keys within the organization, but this requires additional infrastructure and personnel. Another way is to escrow keys to a third party, such as the prevailing Cloud Access Security Broker (CASB) [3], which is a service that provides key management and unified cloud data access control.

Despite all the efforts to encrypt data in the cloud, there are still risks that make us have to strike a balance. First, encryption can be done at different layers and granularities, such as volume-level, object-level, file-level, application-level, and so forth [42]. For the performance reason, it is difficult to implement strong encryption at all layers. As an example, despite implementing volume-level encryption, which is used to be connected to a virtual machine (VM) instance, it is still vulnerable if an attacker gains access to the VM instance. Second, IT administrators or security staff may be necessary to access other personnel's cryptographic keys for key recovery or other reasons. If a disgruntled employee gets the key, it will increase the risk of unauthorized access. This is also an ethical issue. Third, despite not a good practice, CSCs, for technical or budgetary reasons, also escrow keys to the same CSP that also stores the organization's data. This risk of dependency is also termed *Lock-In*, which will be discussed in Sect. 3.2. Finally, due to legal and regulatory requirements for specific encryption algorithms or methods, there is a security gap between different jurisdictions where data has transborder exchange. This issue will be detailed in Sect. 3.3.

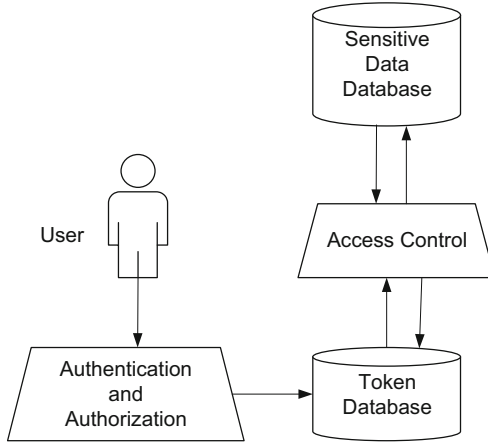


Fig. 2. Overview of tokenization mechanism.

2) Data Obfuscation and De-identification

Concerning security and privacy, practical cloud data protection is necessary to obscure sensitive data or instead use a representation of that data. **Masking** is an elementary data hiding technique (e.g., showing only the last four digits of a credit card number), and similar techniques include *randomization* which replaces part of the data with random characters, and *shuffling* that represents the data with different records within the same dataset. **Tokenization** is another privacy protection technique which is illustrated in Fig. 2, a nonsensitive tag called a token is created as a substitute to be used in place of sensitive data. The implementation of tokenization typically consists of two databases, one storing actual and real sensitive data, and the other storing tokens corresponding to each data entry. A user who needs to access data first obtains nonsensitive tokens, and then a strong access control mechanism such as Identity and Access Management (IAM) [15] decides whether this user can access the corresponding sensitive data entries. **Anonymization** is the primary technique for de-identifying when the data contains Personally Identifiable Information (PII). This process includes removing *direct identifiers* e.g. names, bank accounts, and *indirect identifiers* which are often statistical or demographic information but can be combined to infer PII e.g. personal age and shopping history [6, 8, 31, 37].

There are three major risks during the implementation of the aforementioned security and privacy controls. First, the above techniques can perform well for structured data but may present problems on unstructured data that could be located in any media. Although the existing available solution is DLP and continuous monitoring, this is still not enough to address the challenge of sensitive data mining. Second, the tokenization technique depends on the access control mechanism, thus we have to face all the risks, and the human factor is always the most significant risk among those. This is also an ethical dilemma. Last, although most privacy regulations require data anonymization or de-identification for any

PII use outside of live production environments, how to identify indirect identifiers is also a hard nut to crack due to the lack of effective rules to estimate whether the information is an indirect identifier that seems humble but can be combined with other information to infer PII.

3) *Virtualization*

Proverbially, virtualization technology is the cornerstone of cloud computing, which helps the cloud to implement critically acclaimed on-demand services and resource pooling. In a sense, virtualization is also a security control that achieves access control through the isolation of diverse layers. Despite all the convenience, risks need to be considered while protecting cloud data in practice using virtualization. First, since the hypervisor that manages VM instances is the critical component of the virtualization solution, it tends to be attacked. Compromising a VM instance only results in the data breach within the VM guest, thus threat actors may instead attempt to compromise the hypervisor. Because the hypervisor acts as the interface and controller between the virtualized instances and the host resources, exploiting the hypervisor can affect the security of all VM guests [7]. Another risk is guest escape. Weakly designed or configured VM instances or hypervisors may allow users to break restrictions and leave their own VM instances to gain unauthorized access. There are two ways of guest escape, one is lateral movement, that is, unauthorized access from one VM guest to another one, and the other is vertical movement, that is from one VM guest a user obtains the host machine permissions. As a matter of fact, the second way is more harmful to cloud data protection. Finally, since the cloud environment is multi-tenant, we have to deal with data seizure issues. Legal activity may result in the seizure or inspection of the host machine which has hundreds of VM instances belonging to different CSCs by law enforcement agencies or plaintiff attorneys, even if the organization is not the target. Great efforts still need to be made to cope with this problem by both the security community and the judicial community [39].

3.2 Operational Policies

While technical controls have laid the foundation for mitigating cloud risks, security operations in the cloud provide ongoing security and privacy assurance. This section will discuss several key controls in security operations. Due to space reasons, we will not go into all the details here, but more policies of security operations analysis.

1) *Data Classification*

Data identification and classification are the foundation of cloud data protection. All implemented technical and administrative controls determine the level of protection based on the classification of data. Since the organization is constantly creating data in its operations, this is an operational process, aka “Data Discovery” [17]. Typical approaches to data discovery include label-based, metadata-based, and content-based ones. Whether the data is created on-premises or in the cloud, an assistant tool to data classification is DLP, a technology system

designed to identify, inventory, and control the use of data that an organization deems sensitive, regardless of whether it is employees' personal data, such as web browsing history, pending resignation letters, and so forth. In a nutshell, data discovery can sometimes be a "double-edged sword", raising privacy and ethical concerns.

2) *DRM/IRM*

Data is out of the physical hold of the organization in the cloud, thus compensating controls are needed to protect the data during its lifecycle, especially during the use and share phases. DRM aka IRM which is mentioned in Sect. 2.2 is an ideal mechanism [23, 27]. DRM/IRM usually has the following advantages: (1) *persistent protection*, which follows the information it protects, regardless of where it is located; (2) *dynamic policy control*, which allows data owners to modify access control lists (ACLs) and permissions for the protected data under their control; (3) *remote rights revocation*, which the data owner can revoke permissions at any time; (4) *continuous auditing*, that allow for comprehensive monitoring of the access history.

Despite the many advantages of DRM/IRM, leveraging DRM/IRM in the cloud still faces some challenges. One is *replication restrictions*. DRM/IRM involves permissions for replication and sharing, but the administrative process in the cloud environment often requires creating, shutting down, moving, and backing up VM instances, which is undoubtedly in conflict with the policies of DRM/IRM. The other is *jurisdictional conflicts*. The blurred physical interface brought about by cloud computing will bring about the transborder flow or even out control of a large amount of data, which will lead to regulatory restrictions in different jurisdictions.

3) *Continuous Monitoring*

Automated or continuous monitoring and reporting is an important mechanism for cloud computing to achieve its capability of self-service. The monitoring objects mainly include: (1) *physical environment*, involving the temperature, moderation, and so forth of the data center; (2) *host-level*, including the performance and event tracing of the operating system, middleware, and applications; (3) *network-level*, refers to monitoring various network components, not only hardware and software but also cabling, Software Defined Network (SDN), and control plane.

Continuous monitoring can improve performance and enhance security, however, it can also raise privacy and ethical issues. As an example, the CSP collects the event tracking log of the operating system through the agent installed in the VM instance, so as to obtain the VM guest status and implement anomaly detection of the system. Although the system event log does not contain direct identifiers i.e. PII, the user's behavioral characteristics can still be analyzed by reasoning about system events. Thus those auditing data can be used for precision marketing, and in the worst-case obtained by cyber threat actors to understand user behavior so that they can prepare proper attack vectors.

3.3 Legal and Regulatory Compliance - LRC

Since the essence of cloud computing is to drive business improvement, many of its features such as decentralization and multi-tenancy make it difficult to comply with existing data privacy protection and other laws and regulations.

1) *eDiscovery*

eDiscovery refers to the process of identifying and obtaining electronic evidence for either prosecutorial or litigation purposes. Since cloud computing is often multi-tenant, it is more difficult to find data owned by one CSC without invading data from other CSCs that may reside on the same storage volume, drive, or physical machine. In addition, from a judicial point of view, all evidence needs to be tracked and monitored from the time it is recognized as evidence and acquired for that purpose, which is also called *chain of custody*. While the design of cloud computing may dynamically allocate and recycle resources for other tenants in the same storage location, which is in conflict with judicial principles. Thus when creating security and privacy policies for maintaining a chain of custody or conducting activities requiring the preservation and monitoring of evidence, we need to comply with the regulations.

2) *Diverse Jurisdictions*

A great deal of the difficulties in compliance with the legal and regulation of cloud computing stems from the design of cloud computing. They are often dispersed, often across the county, state, and even international borders. As mentioned earlier, transborder transfer of data is the most difficult reason for cloud to comply with laws and regulations. The governance of compliance must take all of the applied laws and regulations into account to operate reasonably with an understanding of legal risks and liabilities in the cloud.

4 Empirical Evaluation Model

This section will detail our proposed evaluation model for cloud data protection. By analyzing the aforementioned factors that affect cloud data security and privacy, we present a novel algorithm to quantitatively calculate the score of cloud data protection in a specific organization, and show its overall protection level.

4.1 Important Factors

We consider four factors to be important when assessing the protection level of cloud data in an organization.

- **Intra-phase.** As mentioned in Sect. 2.2, a variety of security and privacy controls are implemented at each phase of the cloud data lifecycle. In Table 1 we can see, even within one phase, there may be multiple data states, which means that more controls need to be implemented so that data in different states are protected. To this end, the more states of the data within a phase,

the more assessment is required. Furthermore, data in transit is generally more vulnerable to compromise than data in storage and at rest. Due to the fact that data in transit on public carriers is beyond the confines of the cloud itself. Based on this intuition, we prefer to assign a higher weight to phases with multiple states or containing the transit state in our evaluation model.

- **Inter-phase.** Over time, the more phases cloud data passes through in its lifecycle, the more opportunities it has to be processed, used, and shared. In other words, data in later phases are more critical than in former ones, thus more controls need to be implemented in later phases. As an extreme example, if the data is not properly destroyed during the destruction phase, such as using an insecure method (e.g., simply using the OS *delete* or *rm* command), or if the encryption key is leaked despite leveraging the recommended method of crypto shredding, the organization will eventually lose control of the data that they think this should be destroyed but can still be accessed. In short, the later phases of the evaluation deserve more attention.
- **Lifecycle operations.** Cloud security operations are the guarantee to manage and mitigate cloud data risks within an acceptable range. Although many aspects of security operations are handled by CSPs, that is to say, it is invisible and imperceptible to the CSCs, we should understand that cloud security operations include many global security controls during the whole cloud data lifecycle, such as BC/DR and continuous monitoring mentioned in Sect. 3.2. Our insight is that the cloud security operations should be considered as security and privacy measures applied to each phase of the cloud data lifecycle, and therefore should be given global priority and proper weight in the evaluation model.
- **Compliance.** Compliance requirements are an important aspect of Information Security Management System (ISMS) [2, 5], and there is no doubt that the use of cloud computing presents many challenges in identifying and complying with compliance in specific jurisdictions. In typical cloud scenarios, since resources are allocated dynamically, CSCs do not know the exact physical location of the data. In fact, even CSPs may not know which location the data is in at all times when they manage the VM images and other data, depending on the level of automation and data center design. Similar to security operations, compliance requirements should also be fully evaluated in the assessment model as a global factor.

Hence, to conduct a comprehensive evaluation of cloud data protection, we need to first calculate the scores for intra-phase, inter-phase, operations and compliance, respectively.

4.2 Quantitative Analysis

First, to calculate the score of intra-phase, we define the weights based on different data states within a phase, as shown in Table 3. We prioritize the three data states of in transit, at rest, and in use according to its possibility of risk occurring discussed in Sect. 4.1, and construct a binary truth table. In the last

Table 3. Intra-phase weight rating scale

Phases	Data states			Weight
	In transit	At rest	In use	
CREATE	×	×	×	7 (111)
STORE		×		2 (010)
USE	×	×	×	7 (111)
SHARE	×		×	5 (101)
ARCHIVE	×	×	×	7 (111)
DESTROY		×	×	3 (011)

Table 4. Severity levels of possible risks rating scale

Severity level	Quantitative range	Rounded up average value
Low	0.1–3.9	2.0
Medium	4.0–6.9	6.0
High	7.0–8.9	8.0
Critical	9.0–10.0	10.0

column of Table 3, we can see the weight values generated in different phases due to the existence of different data states, which in parentheses is the binary representation. We further use the weight of intra-phase to compute its protection score, which is defined as:

$$IntraS = \frac{1}{6} \times \left(\sum_{i=1}^6 w_i \times \frac{1}{\max(r_i)} \right) \quad (1)$$

where w_i is the weight of i phase of the cloud data lifecycle defined in Table 3, and r_i means the severity levels of possible risk in the i phase, which can be found in industry standards or best practices. Since the possible risk within the phase with diverse data states is usually technical issues, we identify the risks and assign their severity levels based on the Common Attack Pattern Enumeration and Classification (CAPEC) list defined by US-CERT and DHS with the collaboration of MITRE [35]. For quantitative calculation, we map the severity level to a numerical value based on the conversion table shown in Table 4 included in the Common Vulnerability Scoring System (CVSS) [19]. Thus, we select the highest value of identified risk and then obtain the intra-phase score.

Next, we define the formula for calculating the score of protection level of inter-phase.

$$InterS = \frac{1}{6} \times \left(\sum_{i=1}^6 \frac{1}{(\max(r_i))^{w'_i}} \right) \quad (2)$$

where $w'_i = (10+i)/10$ for the phase of cloud data lifecycle i , which takes into account slightly higher weights for later phases mentioned in Sect. 4.1. Similar

to the formula of intra-phase score, r_i also means the possible risks during the whole lifecycle of cloud data. We use CAPEC and Cloud Controls Matrix (CCM) published by Cloud Security Alliance (CSA) [9] to identify more possible risks. Noting that our evaluation model has the capability of customization mechanism, whereby risk identification model and severity level can be substituted to expert-specified others. Thus, the risk of the highest severity value is identified to represent the most critical risk in each phase and calculate the sum as the inter-phase score.

Then, to calculate lifecycle operations score, we define the formula as:

$$OpS = \frac{1}{\max(r)} \quad (3)$$

where r is the possible operations risk, e.g., data breaches. Similarly, the risk of the highest severity value is the representation of the protection level of operations.

Similar to the *Ops*, the score of compliance is defined as:

$$ComS = \frac{1}{\max(r)} \quad (4)$$

where r is the possible compliance risk based on the organization's geographic location, its jurisdiction and industry. As an example, a bank located in EU needs to comply with GDPR [36] and PCI DSS [11].

Last, we give the overall formula to compute the protection level score of cloud data in an organization, which is defined as:

$$S = \alpha \times IntraS + \beta \times InterS + \gamma \times OpS + \delta \times ComS \quad (5)$$

where the parameters α, β, γ , and δ can be configured based on the expert knowledge and specific application scenarios. Based on our empirical experience, the default values of those parameters are 0.2, 0.2, 0.3, and 0.3 respectively.

4.3 Case Study

We leverage the protection score presented above to assess a financial enterprise that would like to be anonymous, and the result is in accordance with a parallel manual evaluation by experts. Table 5 shows four sub-scores which reflected four evaluation factors mentioned before. For intra-phase score, we obtained the highest severity value in the *SHARE* stage, this is because the target of evaluation has a weak access control for user PII. While for inter-phase score, we identified the highest severity value when data from the *ARCHIVE* stage to *DESTROY* stage due to a lack of approved and unified mechanism for destroying no longer retention data. Then we use the default parameters mentioned in Sect. 4.2, the overall protection score of the target of evaluation's cloud data is 0.46. We mapped this score value to the magnitude of the numeric interval listed in Table 4, and it shows that the overall protection level is medium. This is consistent with the manual qualitative assessment by another team.

Table 5. Case study scores.

	Phases	Severity value of highest risk
IntraS	CREATE	8.00
	STORE	6.00
	USE	2.00
	SHARE	10.00
	ARCHIVE	6.00
	DESTROY	6.00
	Score	6.88
InterS	CREATE	6.00
	STORE	2.00
	USE	6.00
	SHARE	2.00
	ARCHIVE	8.00
	DESTROY	2.00
	Score	1.42
OpS	-	2.00
ComS	-	8.00
Overall Score	-	0.46

5 Related Work

5.1 Cloud Security Assessment

Traditional information system risk assessment mechanisms are still effective for cloud computing environments. However, as a popular computing architecture, the cloud computing environment has some aspects that are unique to other IT system risk assessments. First, the cloud environment involves more entities, including CSPs, CSCs, cloud users, cloud auditors, cloud carriers, and so forth. These stakeholders bring more challenges to cloud security assessment. Second, the technology stack of cloud computing architecture is more complex, and the evaluation targets include components owned and used by multiple parties such as physical environment, virtualization, and applications. In addition, compliance with the cloud environment is also an important aspect of cloud risk assessment [9].

5.2 Data Security and Privacy

Data security, privacy, and ethics come to be widely considered in the security community and among the legal profession. Whether data incorporates security and privacy controls in its life cycle is a critical observation for data security

assessment. Data classification and access control matrix are important ideal data risk assessment tools as well as data security controls. Moreover, continuous monitoring is also used to assess whether the exchange of data violates the organization's data security policy [12–14, 26, 43].

6 Conclusion

In this paper, we make a comprehensive review of each aspect of cloud data protection including security, privacy, and ethical considerations. To evaluate an organization's cloud data protection level, we propose an empirical model that calculates the protection score based on four important factors we consider. A novel algorithm we present can improve the ability of automated evaluation and the credibility of evaluation results. However, frankly speaking, our evaluation model is still a semi-automatically model that also needs experts to identify risks and conduct other manual activities. This will be our further research goal.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is partially supported by CNCERT/CC and (ISC)² Chengdu Chapter.

References

1. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. *ACM Comput. Surv. (CSUR)* **51**(4), 1–35 (2018)
2. Achmadi, D., Suryanto, Y., Ramli, K.: On developing information security management system (isms) framework for ISO 27001-based data center. In: 2018 International Workshop on Big Data and Information Security (IW BIS), pp. 149–157. IEEE (2018)
3. Ahmad, S., Mehruz, S., Beg, J.: Enhancing security of cloud platform with cloud access security broker. In: Kaiser, M.S., Xie, J., Rathore, V.S. (eds.) *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. LNNS, vol. 190, pp. 325–335. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0882-7_27
4. Al-shammari, M.M., Alwan, A.A.: Disaster recovery and business continuity for database services in multi-cloud. In: 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–8. IEEE (2018)
5. Arafat, M.: Information security management system challenges within a cloud computing environment. In: *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pp. 1–6 (2018)
6. Bajaj, P., Arora, R., Khurana, M., Mahajan, S.: Cloud security: the future of data storage. In: Khanna, K., Estrela, V.V., Rodrigues, J.J.P.C. (eds.) *Cyber Security and Digital Forensics. LNDECT*, vol. 73, pp. 87–98. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-3961-6_9
7. Barrowclough, J.P., Asif, R.: Securing cloud hypervisors: a survey of the threats, vulnerabilities, and countermeasures. In: *Security and Communication Networks 2018* (2018)

8. Chen, W.Y., Yu, M., Sun, C.: Architecture and building the medical image anonymization service: cloud, big data and automation. In: 2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), pp. 149–153. IEEE (2021)
9. Cloud Security Alliance (CSA): Cloud Controls Matrix (CCM). <https://cloudsecurityalliance.org/research/cloud-controls-matrix/>
10. Coalfire: Cloud Security Intelligence Report. <https://www.coalfire.com/Documents/Whitepapers/Securealities-Cloud-Security-Report>
11. Council, P.S.S.: Payment Card Industry Data Security Standard. <https://www.pcisecuritystandards.org/>
12. Dahake, S., Chirchi, E.: Maintaining security of the data over cloud: a review. *Int. J. Recent Adv. Multidiscipl. Top.* **2**(4), 4–11 (2021)
13. Darwazeh, N.S., Al-Qassas, R.S., AlDosari, F., et al.: A secure cloud computing model based on data classification. *Proc. Comput. Sci.* **52**, 1153–1158 (2015)
14. Deloitte: Data privacy in the cloud: Navigating the new privacy regime in a cloud environment. <https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/risk/ca-en-risk-privacy-in-the-cloud-pov.PDF>
15. Deochake, S., Channapattan, V.: Identity and access management framework for multi-tenant resources in hybrid cloud computing. arXiv preprint [arXiv:2203.11463](https://arxiv.org/abs/2203.11463) (2022)
16. Dotson, C.: *Practical Cloud Security: A Guide For Secure Design and Deployment*. O'Reilly Media, Sebastopol (2019)
17. Fernandez, R.C., Abedjan, Z., Koko, F., Yuan, G., Madden, S., Stonebraker, M.: Aurum: a data discovery system. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1001–1012. IEEE (2018)
18. Fife, L., Kraus, A., Lewis, B.: *The Official (ISC)² CCSP CBK Reference*. John Wiley & Sons, New York (2021)
19. FIRST: Common vulnerability scoring system v3.0: Specification document. <https://www.first.org/cvss/specification-document>
20. Fortinet: Cloud Security Report. <https://www.fortinet.com/content/dam/fortinet/assets/analyst-reports/ar-cybersecurity-cloud-security.pdf>
21. Fugkeaw, S., Worapaluk, K., Tuekla, A., Namkeatsakul, S.: Design and development of a dynamic and efficient PII data loss prevention system. In: Meesad, P., Sodsee, S., Jitsakul, W., Tangwannawit, S. (eds.) *IC2IT 2021*. LNNS, vol. 251, pp. 23–33. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79757-7_3
22. Gunleifsen, H., Kemmerich, T., Gkioulos, V.: Dynamic setup of IPSEC VPNS in service function chaining. *Comput. Netw.* **160**, 77–91 (2019)
23. Hussain, A., Kiah, M.L.M., Anuar, N.B., Md Noor, R., Ahmad, M.: Performance and security challenges digital rights management (DRM) approaches using fog computing for data provenance: a survey. *J. Med. Imaging Health Inform.* **10**(10), 2404–2420 (2020)
24. International Standards Organization/International Electrotechnical Commission: ISO/IEC 17789:2014 Information technology - Cloud computing - Reference architecture. <https://www.iso.org/standard/60545.html>
25. (ISC)²: Cloud Security Report 2021. <https://www.isc2.org/-/media/ISC2/Research/Resource-Thumbnails/Resource-Center/Research/2021-Cloud-Security-Report-FINAL.ashx>
26. Kacha, L., Zitouni, A.: An overview on data security in cloud computing. In: Silhavy, R., Silhavy, P., Prokopova, Z. (eds.) *CoMeSySo 2017*. AISC, vol. 661, pp. 250–261. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67618-0_23

27. Ko, H., Měsíček, L., Choi, J., Hwang, S.: A study on secure medical-contents strategies with DRM based on cloud computing. *J. Healthcare Eng.* **2018**, 6410180 (2018)
28. Krishnan, S., Neyaz, A., Shashidhar, N.: A survey of security and forensic features in popular eDiscovery software suites. *International Journal of Security (IJS)* **10**(2), 16 (2019)
29. Kutame, F.N., Ochara, N.M., Kadyamatimba, A., Sotnikov, A., Fiodorov, I., Telnov, Y.: A case study of cloud-based business continuity model. In: *CEUR Workshop Proceedings*, pp. 26–35 (2021)
30. Lopez, J., Oppliger, R., Pernul, G.: Authentication and authorization infrastructures (AAIS): a comparative survey. *Comput. Secur.* **23**(7), 578–590 (2004)
31. Madavi, K.B., Karthick, P.V.: Enhanced cloud security using cryptography and steganography techniques. In: *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1, pp. 90–95. IEEE (2021)
32. Mei, R., Yan, H.-B., Han, Z.-H.: RansomLens: understanding ransomware via causality analysis on system provenance graph. In: Lu, W., Sun, K., Yung, M., Liu, F. (eds.) *SciSec 2021. LNCS*, vol. 13005, pp. 252–267. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89137-4_18
33. Mei, R., Yan, H.B., Han, Z.H., Jiang, J.C.: CTScopy: hunting cyber threats within enterprise via provenance graph-based analysis. In: *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pp. 28–39. IEEE (2021)
34. Mei, R., Yan, H., Wang, Q., Han, Z., Lyu, Z.: TDLens: toward an empirical evaluation of provenance graph-based approach to cyber threat detection. *China Commun.* **19**, 102–115 (2022)
35. MITRE: CAPEC: Common Attack Pattern Enumeration and Classification. <https://capec.mitre.org/index.html>
36. Parliament, E.: General Data Protection Regulation. <https://gdpr-info.eu/>
37. Patil, S., Joshi, S., Patil, D.: Enhanced privacy preservation using anonymization in IOT-enabled smart homes. In: Satapathy, S.C., Bhateja, V., Mohanty, J.R., Udgata, S.K. (eds.) *Smart Intelligent Computing and Applications. SIST*, vol. 159, pp. 439–454. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-9282-5_42
38. Shukla, M.K., Dubey, A.K., Upadhyay, D., Novikov, B.: Group key management in cloud for shared media sanitization. In: *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 117–120. IEEE (2020)
39. Srivastava, P., Choudhary, A.: Evolving evidence gathering process: cloud forensics. In: Tiwari, S., Suryani, E., Ng, A.K., Mishra, K.K., Singh, N. (eds.) *Proceedings of International Conference on Big Data, Machine Learning and their Applications. LNNS*, vol. 150, pp. 227–243. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8377-3_20
40. du Toit, J.: Digital rights management to protect private data on the internet. In: *ECCWS 2018 17th European Conference on Cyber Warfare and Security V2*, p. 128. Academic Conferences and Publishing Limited (2018)
41. Wang, Q., Yan, H., Han, Z.: Explainable apt attribution for malware using NLP techniques. In: *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pp. 70–80. IEEE (2021)
42. Yang, P., Xiong, N., Ren, J.: Data security and privacy protection for cloud storage: a survey. *IEEE Access* **8**, 131723–131740 (2020)

43. Zhang, J., Chen, B., Zhao, Y., Cheng, X., Hu, F.: Data security and privacy-preserving in edge computing paradigm: survey and open issues. *IEEE Access* **6**, 18209–18237 (2018)
44. Zhang, Z., Cheng, Y., Gao, Y., Nepal, S., Liu, D., Zou, Y.: Detecting hardware-assisted virtualization with inconspicuous features. *IEEE Trans. Inf. Forensics Secur.* **16**, 16–27 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Anomaly Detection



A Self-supervised Adversarial Learning Approach for Network Intrusion Detection System

Lirui Deng¹, Youjian Zhao^{1,2(✉)}, and Heng Bao³

¹ Department of Computer Science and Technology, Tsinghua University, Beijing, China

dlr18@mails.tsinghua.edu.cn, zhaoyoujian@tsinghua.edu.cn

² Zhongguancun Laboratory, Beijing, China

³ School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

baoheng@iie.ac.cn

Abstract. The network intrusion detection system (NIDS) plays an essential role in network security. Although many data-driven approaches from the field of machine learning have been proposed to increase the efficacy of NIDSs, it still suffers from extreme data imbalance and the performance of existing algorithms depends highly on training datasets. To counterpart the class-imbalanced problem in network intrusion detection, it is necessary for models to capture more representative clues within same categories instead of learning from only classification loss. In this paper, we proposed a self-supervised adversarial learning approach for intrusion detection, which utilize instance-level discrimination for better representation learning and employs a adversarial perturbation styled data augmentation to improve the robustness of NIDS on rarely seen attacking types. State-of-the-art result was achieved on multiple frequently-used datasets and experiment conducted on cross-dataset setting demonstrated good generalization ability.

Keywords: Network intrusion detection · Self-supervise learning · Adversarial learning

1 Introduction

While the advent of the Internet has brought immense convenience to our daily lives in recent decades, it has also unavoidably introduced dozens of new challenges. As people nowadays spend more time in cyberspace than real world no matter living or working, attacking on network activities with various kinds of intrusion techniques to prey privacy information or corporation confidential information has never stop. Therefore, as a counterpart, the intrusion detection system (IDS) which safeguard the integrity and availability of key assets has always

been a hot research topic in computer and network security community. In contrast to host-based IDS which are distributed at end point users' system, network intrusion detection system (NIDS) primarily characterized as a solution inside the data transfer pipeline between computers that can monitor the network traffic and alert or even take active response measures when malicious behavior is spotted [4]. Other than some NIDS designed for specific network environment [16, 19, 24] like Hadoop-based platforms or particular cloud system, most general NIDS researches [10, 14, 33, 38] were performed on network intrusion detection datasets to demonstrate and compare their effectiveness and generalization ability in a data-driven fashion.

Among several limitations of existing algorithms, data imbalance in different classes, especially the lack of data in rarely seen attacking categories, is a one of the most challenging problems. However, it is also a very common phenomenon in network intrusion detection datasets considering the difficulty in data collection or generation. Benign traffic is no doubt the majority part of internet data transfer, not to mention the inherent nature of malicious network activity as of being disguised. While the performance of most traditional ML-based method declines significantly in the case of learning from imbalanced data, a large amount of researches try to address this problem by various approaches [5, 8, 20, 28, 34, 36, 38]. Recently, contrastive learning has drawn a lot of attention with impressive performance improvement [27, 35] in computer vision and natural language processing. Besides supervised contrastive learning, instance-level discrimination framework in self-supervised fashion have also shown promising result with few-Shot classification [21] and quickly being used in NIDS research [22].

Inspired by the success of contrastive learning and adversarial learning in CV and NLP, in this paper we proposed a self-supervised adversarial learning (SSAL) approach for network intrusion detection. The main contributions of this paper are as follows:

- First, we utilized an adversarial learning approach for NIDS with design of netflow-based adversarial examples, which improves robustness on class-imbalanced datasets by explicitly suppressing the vulnerability in the representation space and maximizing the similarity between clean examples and their adversarial perturbations.
- We proposed a 2-stage pre-train style self-supervised learning in SSAL that leverages instance-level self-supervised contrastive learning and adversarial data augmentation to achieve a better representation over limited sample, which has not been proposed for NIDS to the best of our knowledge.
- We conducted a experimental evaluation with existing methods on multiple datasets including UNSW-NB15 [26] and CIC-IDS-2017/2018 [31], which shows boosted performance of several machine learning baselines across different datasets.

2 Related Work

In this section we summarize the algorithms and research work related to this study.

2.1 Network Intrusion Detection System

Data-driven methods have been developed and deployed for NIDSs for more than two decades [9]. In order to achieve an effective NIDS, various methods including both machine learning (ML) and deep learning (DL) techniques have been proposed by research community.

Traditional machine learning algorithm such as KNN, PCA, SVM, and tree-based models have all been adopted with intrusion detection, and often used as baseline for particular improved module. For example, Gao *et al.* [11] used classification and regression trees (CARTs) on NSL-KDD datasets with a ensemble scheme where multiple trees were trained on adjusted sampling. Karatas *et al.* [17] addressed the dataset imbalance problem by reducing the imbalance ratio using Synthetic Minority Oversampling Technique (SMOTE), and used different ML algorithms as a baseline for cross comparison that shows improved detection ability for minority class attacks.

Recent studies suggested that the use of DL algorithms for NIDSs have much superior performance than the ML-based methods. RNN and autoencoder [1] was pointed to be the most frequently used models for NIDS in past decades. Regarding data imbalance, Yu *et al.* proposed a CNN-based few shot learning model to improve the detection reliability of network attack categories with the few sample problem. Manocchio proposed FlowGAN [23] which utilized generative models for data augmentation. However, most DL schemes are more complex and require extensive computing resources compare to ML-based methods.

2.2 NIDS Datasets

High-quality data sets are definitely required to fully evaluate the performance of various intrusion detection systems. Many contributions have been published in recent years containing representative network flow data with different kinds of preprocess, which are provided mainly in three categories of formats.

Packet Based Data. The most original and commonly used format is packet based data captured in pcap format and contains payload. Early NIDS datasets does not provide packet based data because it takes too much storage space. But datasets published more recently like CIC-IDS-2017/2018, UNSW-NB15 and LITNET-2020 [7] tend to provide both pcap files and flow based features for the benefit of comparison between different NIDS methods.

Flow Based Data. Flow based data is much more condensed compare to packet based data. It aims to describes the behavior of whole network connection session by aggregate all packets sharing same properties within a time window. Commonly used flow-based formats includes NetFlow [6], OpenFlow [25] and NFStream [2]. CICFlowmeter (formerly known as ISCXFlowMeter [32]) is another important network flow format generator, which tranfers pcap files into more than 80 netflow features, since it was published by Canadian Institute for Cybersecurity therefore used by both CICIDS-2017 and CICIDS-2018.

Other Data. This summarize all data sets that are neither purely packet-based nor flow-based. For example, The KDD CUP 1999 [18] contains host-based attributes like number of failed logins, which can only obtained from above network interface. As a consequence, dataset of this category has its own set of attributes and can not be unified with each other.

2.3 Contrastive Learning

Contrastive learning techniques has been widely used in metric learning such as triplet loss [30] and contrastive loss [13]. While in recent self-supervised approaches, contrastive learning mostly shares a core idea of minimizing various kinds of contrastive loss (i.e. NCE [12], infoNCE [27]) evaluated on pairs of data augmentations. Typically, augmentations are obtained by data transformation (i.e. rotation, cropping, color Jittering in CV, or masking in NLP), but using “adversarial augmentations” as challenging training pairs that maximize the contrastive loss shows more robustness in recently study [15].

3 Approach

In this section, we will explain the main algorithms of our proposed self-supervised adversarial learning framework for data imbalance network intrusion detection.

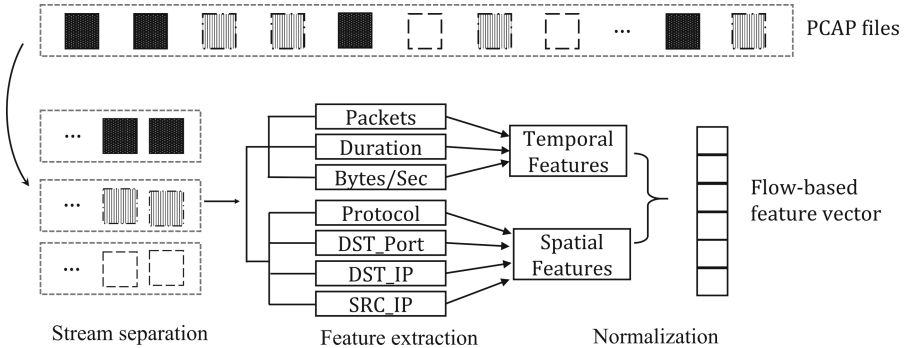


Fig. 1. preprocess pipeline from PCAP files to flow-based feature vector

3.1 Data Preprocessing

To build a comparable cross-dataset evaluation process, we adopt commonly used datasets UNSW-NB15, CIC-IDS-2017 and CIC-IDS-2018, as they not only contain a wide range of attack scenarios but also provide original pcap files that can be easily processed into unified feature set. CIC-IDS-2017 dataset is made up of 5 days network traffic with 7 different network attacking, which forms 51GB

size of data. The benign traffic was generated with profile system to protect user privacy. It provides both network traffic (pcap files) and event logs for attack label on each machine. CIC-IDS-2018 dataset is also created by CICFlowMeter but with both benign and malicious profile system, and has more than 400GB pcap data among 17 days. UNSW-NB15 was release in 2015 by Australian Centre for Cyber Security (ACCS) that contains a total of 100 GB of pcap files, consist of 2,218,761 (87.35%) benign flows and 321,283 (12.65%) attack ones.

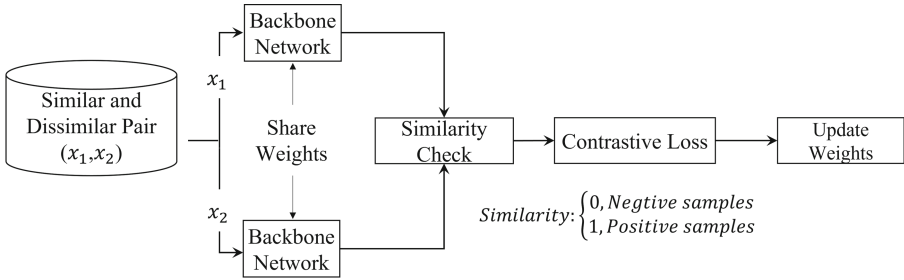
After obtaining original PCAP files, we follow the setting from [29] and take 43 extended feature dimension from the latest *netflow version 9 flow-record format* [6] for flow-based feature extraction (full feature set can be obtained from [29]). Netflow was proposed by Cisco and has become one of the most commonly used flow-based formats for recording network traffic. A network flow stream is an aggregation of a sequence of packets in a continuous session (of TCP connection by default) with the same source IP, source port, destination IP, destination port, and transport protocol. The distribution of our processed unified dataset is shown at Table 1.

Table 1. Distribution of Unified Dataset

	NF-UNSW-NB15	CIC-IDS-2018	CIC-IDS-2017	Summary	Ratio
Benign	2295222	16635567	2359087	21289876	88.29%
Fuzzers	22310	0	0	22310	0.09%
Analysis	2299	0	0	2299	0.01%
Backdoor	2169	0	0	2169	0.01%
DoS	5794	483999	252660	742453	3.08%
Exploits	31551	0	0	31551	0.13%
Generic	16560	0	0	16560	0.07%
Reconnaissance	12779	0	0	12779	0.05%
Shellcode	1427	0	0	1427	0.01%
Worms	164	0	0	164	0.00%
BruteForce	0	120912	15994	136906	0.57%
Bot	0	143097	1966	145063	0.60%
DDoS	0	1390270	41835	1432105	5.94%
Infiltration	0	116361	0	116361	0.48%
Web Attack	0	3502	0	3502	0.01%
portscan	0	0	158930	158930	0.66%

Session stream separation might be a little tricky since streams obtained by only quintuple may not be accurate and contain too much data packets. Inspired by [37], other than following tcp handshake flags, we further segment streams by a timeout mechanism to cut idle stream into more pieces with periodic reset. The procedure of generating NIDS datasets with unified feature set is show in Fig. 1.

3.2 Self-supervised Adversarial Learning



(a) Vanilla Contrastive Learning

(b) Self-supervised Adversarial Learning

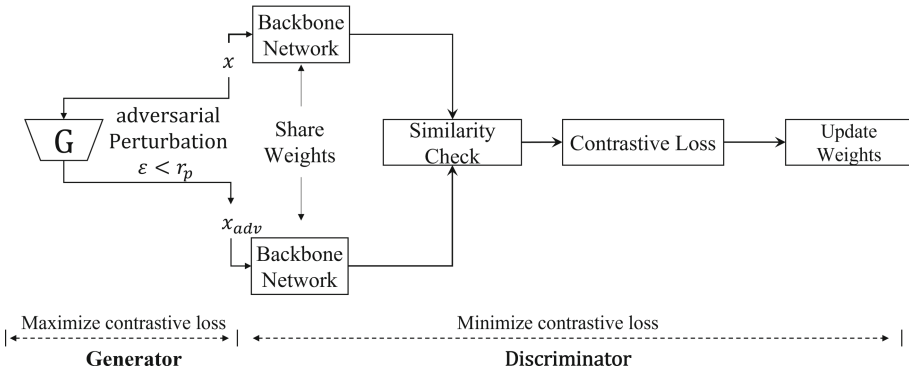


Fig. 2. Self-supervised Adversarial Learning vs. Vanilla Contrastive Learning

In self-supervised styled contrastive learning (CL), the dataset $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^N$ is unlabeled, and each example \mathbf{x}_i from a mini-batch is either paired with a positive sample \mathbf{x}'_i by transformations \mathbf{T} or a negative sample $x_j/x'_j, j \neq i$. CL seeks to learn an invariant representation of \mathbf{x}_i by minimizing the distance between positive samples defined as:

$$\mathcal{L}_{CL} = -\log \frac{\exp(\text{sim}(\mathbf{x}_i, \mathbf{x}'_i))}{\sum \exp(\text{sim}(\mathbf{x}_i, \mathbf{x}_k))} \tag{1}$$

While Chen et al. demonstrate in SimCLR [3] that a temperature parameter τ and a non-linear projector \mathbf{G} after backbone network is crucial to the performance of self-supervised CL, we adopt SimCLR loss $\mathcal{L}_{\text{SimCLR}}$ for the base setting of SSAL:

$$\mathcal{L}_{\text{SimCLR}}(\mathbf{x}_i, \mathbf{x}_j) = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{2}$$

where $\mathbf{h}_i = f(\mathbf{x}_i), \quad \mathbf{h}_j = f(\mathbf{x}_j),$
and $\mathbf{z}_i = g(\mathbf{h}_i), \quad \mathbf{z}_j = g(\mathbf{h}_j)$

Adversarial Attack. The design of positive and negative sampling strategy is key to performance of CL models, and the robustness of model will largely depend on the difficulty of proposed sample pairs. As opposed to vanilla contrastive learning, self-supervised adversarial learning leverages adversarial augmentation to ease the difficulty in hard sample mining. Define the perturbation ϵ using L_∞ -Norm attack for example:

$$\epsilon = \arg \max_{\|\epsilon\|_\infty} \mathcal{L}_{\text{SimCLR}}(\mathbf{x}_i, \mathbf{x}_i + \epsilon) \quad (3)$$

With perturbations ϵ given in certain radius that lead to the most diverse positive pairs, we have a adversarial training scheme by both encouraging the learning algorithm to produce a more invariant representation upon updating parameter θ and then find the ϵ' under θ' again. This pipeline is described in Fig. 2 (Fig. 3).

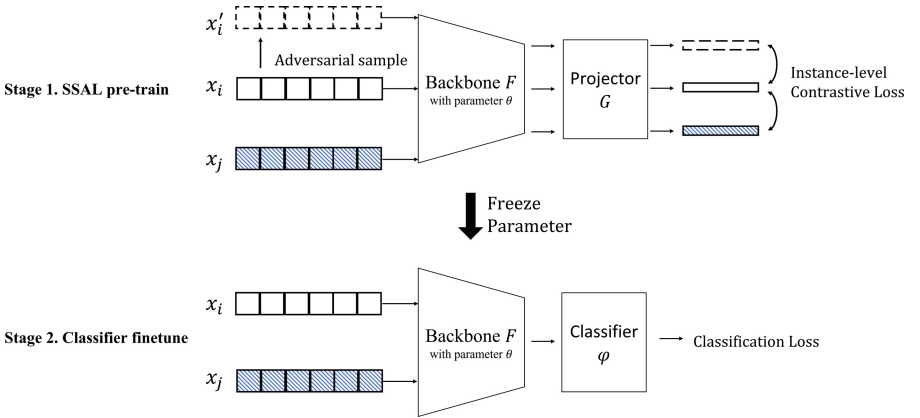


Fig. 3. Framework of proposed 2-stage SSAL NIDS training process

3.3 Classifier Fine-Tune

With SSAL we can already pre-train the model without any class labels in adversarial fashion, but without class annotation pre-trained model cannot be directly used for class-level classification.

Therefore we froze the parameter θ from pre-trained model f , and switch projector head g with a non-linear classifier ψ . The training was conducted under standard multi-class single-label training:

$$\begin{aligned} \mathbf{z}_i &= \psi(f(\mathbf{x}_i)), \quad \text{for } i = 1, 2, \dots, N \\ p_{i,c} &= \sigma(z_{i,c}) = \frac{e^{z_{i,c}}}{\sum_{j=1}^M e^{z_{i,j}}}, \quad \text{for } c = 1, 2, \dots, M \end{aligned} \quad (4)$$

with cross entropy loss:

$$\mathcal{L}_{ce}(\mathbf{x}_i, \mathbf{l}_i) = - \sum_{c=1}^M y_{i,c} \log(p_{i,c}) \quad (5)$$

The full process of proposed 2-stage SSAL for NIDS is shown in Algorithm 1.

Algorithm 1: self-supervised adversarial learning for NIDS

```

1 Stage1 SSAL pre-train
   | input : Dataset  $D = \{\mathbf{x}_i\}_{n=1}^N$ 
   | output: model  $f$ 
2   Initial model  $f$  with parameter  $\theta$  and projector  $g$ 
3   repeat
4     | for all  $x \in \text{minibatch } \mathbf{B}$  do
5       | | generate  $\epsilon = \arg \max_{\|\epsilon\|_\infty} \mathcal{L}_{\text{SimCLR}}(\mathbf{x}_i, \mathbf{x}_i + \epsilon)$ 
6       | |  $\theta' = \theta + \nabla_x \mathcal{L}_{\text{SimCLR}}(\mathbf{x}, \mathbf{x} + \epsilon)$ 
7       | end
8   until reach epoch  $N$  or  $L \leq \delta_1$ 
1 Stage2 Classifier Fine-tune
   | input : Dataset with label  $D = \{\mathbf{x}_i, \mathbf{l}_i\}_{n=1}^N$ , model  $f$  with parameter  $\theta$ 
   | output: model  $f$  and classifier  $\psi$ 
2   Initial classifier  $\psi$  with parameter  $\rho$ , freeze  $\theta$ 
3   repeat
4     | for all  $x \in \text{minibatch } \mathbf{B}$  do
5       | |  $\rho' = \rho + \nabla_x \mathcal{L}_{ce}(\mathbf{x}_i, \mathbf{l}_i)$ 
6       | end
7   until reach epoch  $N$  or  $L \leq \delta_2$ 

```

4 Experiment Results

Metric and Implementation. The evaluation is conducted by comparing the classifier performance with various classification metrics. The intrusion detection datasets we evaluate on contain several attacking categories, which can be treated as both binary classification and multiple classification problem. While comparing performance under binary classification scenario, the basic terms used in the evaluation is as follow:

$$\begin{aligned}
 \text{Accuracy}(ACC) &= \frac{TP + TN}{TP + FP + TN + FN}, \\
 \text{DetectionRate}(DR) &= \frac{TP}{TP + FN}, \quad \text{a.k.a } \text{Recall}, \\
 \text{Precision} &= \frac{TP}{TP + FP}, \\
 \text{F1Score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
 \end{aligned} \quad (6)$$

where TP stands for numbers of true positive samples, FN for false negative, and so forth.

For multi-class classification setting with more detailed label of attacking types, weighted average measure of above metric was adopted considering the proportion for each label in the dataset. To achieve a fair evaluation, five cross-validation splits are conducted and the mean is measured.

Evaluation on Unified Feature Dataset. With the unified feature set upon pre-processed UNSW-NB15 and CIC-IDS-2017/2018 dataset mentioned in Sect. 3.1, we conduct a evaluation across multiple datasets. For the purpose of comparison, we implemented a simple MLP and the Extra Trees model from [29] as baseline models. In Table 2, we can see that our SSAL method achieved outstanding result in all three datasets and exceed previous works in most metrics.

Table 2. Performance on unified dataset

Dataset	Metric	MLP	Extra trees [29]	SSAL
NF-UNSW-NB15	ACC	91.02	99.73	99.71
	DR	79.45	97.07	97.45
CIC-IDS-2017	ACC	88.42	97.46	99.57
	DR	82.61	96.54	97.14
CIC-IDS-2018	ACC	83.13	99.35	99.89
	DR	76.63	97.12	98.63
Overall	ACC	88.1	97.91	99.63
	DR	81.6	96.65	97.35

Table 3 presents the detailed detection results of different attacking class on the merged NIDS dataset. While using the same backbone (Multi-Layer Perceptron), the performance of model with SSAL pre-train was largely improved on rare seen attacking data.

Table 3. Detailed performance of different classes on unified dataset.(ACC)

ClassName	MLP	SSAL	Class Name	MLP	SSAL
Benign	81.53	99.14	Shellcode	21.73	89.13
Fuzzers	64.31	84.65	Worms	34.86	60.91
Analysis	46.82	82.43	BruteForce	65.69	88.54
Backdoor	49.73	85.77	Bot	61.34	81.17
DoS	54.34	97.63	DDoS	53.52	99.76
Exploits	59.11	93.22	Infiltration	54.43	73.82
Generic	58.27	88.61	Web Attack	62.77	71.44
Reconnaissance	32.92	91.28	portscan	46.98	85.38
Overall	76.63	98.63			

Further Ablation. To further demonstrate the superiority of our proposed method, we compare our method with different backbone networks with ablation studies upon SSAL modules. We first use two different frequently used backbones, MLP and CNN, and plug them with SSAL pre-train for representation learning. The evaluation result shown on Table 4 proves that SSAL can effectively enhance the ability of network intrusion detection systems. As for feature extraction, Table 5 shows the result of different classifiers when SSAL was used as a feature extractor. We first pre-train with all unlabeled training data with SSAL for feature extraction, then freeze the network parameter and use SVM or k-NN as a classifier to check the representative ability of SSAL model.

Table 4. Performance with different backbone.(ACC)

ACC	UNSW-NB15	CIC-IDS-2017
MLP	87.81	88.63
MLP + SSAL	98.37	98.82
CNN	91.44	90.52
CNN + SSAL	97.53	97.74

Table 5. Performance with different classifier.(ACC)

CIC-IDS-2017	SVM	k-NN
w/o SSAL	85.62	81.46
with SSAL	96.72	94.91

5 Conclusion and Discussions

In this paper, we try to tackle the data imbalance problem in network intrusion detection with adversarial style data augmentation and self-supervised contrastive representation learning. More specifically, we proposed a self-supervised adversarial learning way to enhance the representative learning progress in deep learning based NIDS, which utilizing a instance-wise attack to yield a robust model by suppressing their adversarial vulnerability against perturbation samples. State-of-the-art performance was achieved on commonly used Experiments on multiple datasets show improvement of proposed learning framework against vanilla DL approach with same backbones.

In addition to the conclusion, there are also some works could be done in the future. Although we among other researchers have made a lot of effort on data imbalance for network intrusion detection problems, there are still more gaps need to be filled to a robust and applicable NIDS. For instance, in our method the result from different feature sets shows noticeable performance gap. we believe that to further improve the representative ability of network flow

data with a standard and comprehensive behavior feature set is key to better data-driven NIDS solution. Also we are looking forward to explore an universal end-to-end approach for more generalized NIDS which could greatly reduce the difficulty of system deployment.

References

1. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **32**(1), e4150 (2021)
2. Aouini, Z., Pekar, A.: Nfstream: a flexible network data analysis framework. *Computer Networks*, p. 108719 (2022)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*, pp. 1597–1607. PMLR (2020)
4. Chou, D., Jiang, M.: A survey on data-driven network intrusion detection. *ACM Comput. Surv.* **54**(9), 1–36 (2021)
5. Chowdhury, M.M.U., Hammond, F., Konowicz, G., Xin, C., Wu, H., Li, J.: A few-shot deep learning approach for improved intrusion detection. In: *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 456–462. IEEE (2017)
6. Claise, B.: Cisco systems netflow services export version 9. RFC **3954**, 1–33 (2004)
7. Damasevicius, R., et al.: Litnet-2020: an annotated real-world network flow dataset for network intrusion detection. *Electronics* **9**(5), 800 (2020)
8. Ding, H., Chen, L., Dong, L., Fu, Z., Cui, X.: Imbalanced data classification: a KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* **131**, 240–254 (2022)
9. Dokas, P., Ertöz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.N.: Data mining for network intrusion detection. In: *Proceedings of the NSF Workshop on Next Generation Data Mining*, pp. 21–30. Citeseer (2002)
10. Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H.: Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *J. Inform. Secur. Appl.* **50**, 102419 (2020)
11. Gao, X., Shan, C., Hu, C., Niu, Z., Liu, Z.: An adaptive ensemble machine learning model for intrusion detection. *IEEE Access* **7**, 82512–82521 (2019)
12. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings (2010)
13. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, vol. 2, pp. 1735–1742. IEEE (2006)
14. Hindy, H., et al.: Leveraging siamese networks for one-shot intrusion detection model. *arXiv preprint arXiv:2006.15343* (2020)
15. Ho, C.H., Nvasconcelos, N.: Contrastive learning with adversarial examples. *Adv. Neural Inform. Process. Syst.* **33**, 17081–17093 (2020)
16. Jeong, H.D.J., Hyun, W., Lim, J., You, I.: Anomaly teletraffic intrusion detection systems on hadoop-based platforms: a survey of some problems and solutions. In: *2012 15th International Conference on Network-Based Information Systems*, pp. 766–770. IEEE (2012)

17. Karatas, G., Demir, O., Sahingoz, O.K.: Increasing the performance of machine learning-based IDSS on an imbalanced and up-to-date dataset. *IEEE Access* **8**, 32150–32162 (2020)
18. Kdd cup 1999: Computer network intrusion detection (1999). <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
19. Keegan, N., Ji, S.-Y., Chaudhary, A., Concolato, C., Yu, B., Jeong, D.H.: A survey of cloud-based network intrusion detection analysis. *Hum. Centric Comput. Inform. Sci.* **6**(1), 1–16 (2016). <https://doi.org/10.1186/s13673-016-0076-z>
20. Lee, J., Park, K.: Gan-based imbalanced data intrusion detection system. *Person. Ubiquitous Comput.* **25**(1), 121–128 (2021)
21. Liu, C., et al.: Learning a few-shot embedding model with contrastive learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 8635–8643 (2021)
22. Liu, L., Wang, P., Ruan, J., Lin, J.: Conflow: contrast network flow improving class-imbalanced learning in network intrusion detection. *Research Square Preprint* (2022)
23. Manocchio, L.D., Layeghy, S., Portmann, M.: Flowgan-synthetic network flow generation using generative adversarial networks. In: *2021 IEEE 24th International Conference on Computational Science and Engineering (CSE)*, pp. 168–176. IEEE (2021)
24. Manzoor, M.A., Morgan, Y.: Real-time support vector machine based network intrusion detection system using apache storm. In: *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–5. IEEE (2016)
25. McKeown, N., et al.: Openflow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **38**(2), 69–74 (2008)
26. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6. IEEE (2015)
27. Van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. *arXiv e-prints* pp. arXiv-1807 (2018)
28. Pan, T., Zhao, J., Wu, W., Yang, J.: Learning imbalanced datasets based on smote and gaussian distribution. *Inform. Sci.* **512**, 1214–1233 (2020)
29. Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks Appl.* **27**(1), 357–370 (2022)
30. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. *Adv. Neural Inform. Process. Syst.* **16** (2003)
31. Sharafaldin, I., Gharib, A., Lashkari, A.H., Ghorbani, A.A.: Towards a reliable intrusion detection benchmark dataset. *Softw. Network.* **2018**(1), 177–200 (2018)
32. Shiravi, A., Shiravi, H., Tavallae, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **31**(3), 357–374 (2012). <https://doi.org/10.1016/j.cose.2011.12.012>, <https://www.sciencedirect.com/science/article/pii/S0167404811001672>
33. Thomas, R., Pavithran, D.: A survey of intrusion detection models based on NSL-KDD data set. In: *2018 Fifth HCT Information Technology Trends (ITT)*, pp. 286–291 (2018)
34. Wang, T., Lv, Q., Hu, B., Sun, D.: A few-shot class-incremental learning approach for intrusion detection. In: *2021 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–8. IEEE (2021)

35. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)
36. Xu, C., Shen, J., Du, X.: A method of few-shot network intrusion detection based on meta-learning framework. *IEEE Trans. Inform. Foren. Secur.* **15**, 3540–3552 (2020)
37. Yu, L., et al.: PBCNN: packet bytes-based convolutional neural network for network intrusion detection. *Comput. Networks* **194**, 108117 (2021)
38. Zhang, H., Huang, L., Wu, C.Q., Li, Z.: An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **177**, 107315 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Anomaly Detection of E-commerce Econnoisseur Based on User Behavior

Yangyu Long, Wei Zhao, Jilong Yang, Jincheng Deng^(✉), and Fangming Liu

Beijing Knownsec Information Technology Co., Ltd., Beijing 100097, China
{longyy3, zhaow, yangjl, dengjc, liufm}@knownsec.com,
jcdeng_209@163.com

Abstract. Econnoisseur refers to users who obtain high returns from the Internet at low cost. It is of great significance for platform to identify econnoisseur to reduce unnecessary losses. At present, econnoisseur is mainly intercepted by rules. This method will fail when the new get the best deal method appears, and there is a certain lag. This paper identifies the econnoisseur from Knownsec Security Intelligence Brain's e-commerce website visitors. First of all, it is found that the precision and recall of the Isolation Forest are better than the Local Outlier Factor and DBSCAN in econnoisseur detection. Secondly, we merged the similar URLs visited by users with Bi-directional Long Short-Term Memory (BiLSTM), then use the merged data in Isolation Forest Model. It is found that the improved Isolation Forest model based on BiLSTM can further improve the detection ability. Practical case studies showed that this method has certain validity and reference for the detection of econnoisseur.

Keywords: Econnoisseur · Isolated Forest · Local Outlier Factor · Anomaly detection · Bidirectional Long Short Term Memory · User behavior

1 Introduction

In recent years, e-commerce platforms have shown a trend of peak traffic dividends. Each platform will provide marketing activities to win customers and improve user stickiness. The process also give birth to the econnoisseur who exploit vulnerabilities in platform activity to profit.

According to the analysis report on the application of digital financial anti fraud Technology (2021) released by Institute of Cloud Computing and Big Data of China Academy of Information and Communications Technology and the ICBC Security Attack and Defense Laboratory, it is found that the total loss caused by the anti-fraud of black industry(see Fig. 1), showing an increasing trend every year, and the loss is expected to reach 710 billion yuan in 2022, the econnoisseur accounts for a large proportion. In 2019, Pinduoduo was robbed of tens of millions of yuan by the econnoisseur within a few hours because of an expired coupon bug on the platform. In 2021, Jingdong Mall was discovered and spread by the econnoisseur due to the wrong coupon setting, resulting in a direct loss of nearly 70 million yuan. On the one hand, the existence of the econnoisseur damages the profits of ordinary users, on the other hand, it also greatly reduces the company's activities, and its governance is urgent.

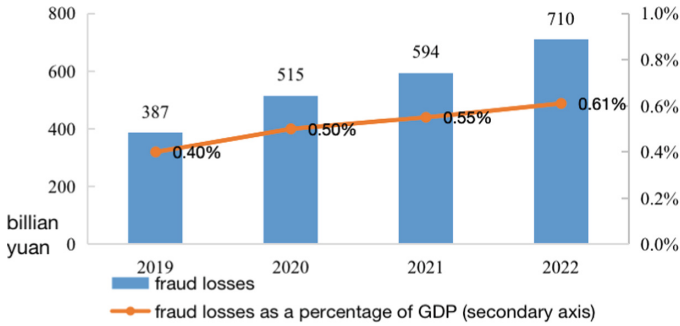


Fig. 1. Fraud losses and forecasts as a percentage of GDP

2 Related Work

The application of anomaly detection in the field of cyber security focuses on APT detection, intrusion detection and so on.

APT attack is a hidden and persistent network intrusion process, which carries out advanced persistent threats against specific targets. Bohara [1] compared various unsupervised algorithms such as K-means for APT detection and found that it can detect infected hosts. Zhong Yao [2] performed anomaly detection on traffic log data based on Isolated Forest and found that it has certain detection ability against APT attacks and can mark the suspected infected hosts.

Intrusion detection is a system that detects intruders in a network. The detection methods can be divided into supervised and unsupervised based machine learning algorithms. References [3–6] are mainly based on supervised algorithms such as Naive Bayes, Bayesian Networks, Hidden Markov Models, and ensemble learning for intrusion detection. This detection method requires a large number of labeled sample data, but there will be insufficient sample label data in many scenarios. In literature [7–11], unsupervised algorithms such as K-means clustering, hierarchical clustering and DBSCAN are used for intrusion detection, which has good detection ability.

At present, the research of econnoisseur detection is still in the theoretical stage, the engineering is mainly based on traditional threshold setting or rule-based interception. Yuan Dandan [12] based on the community discovery algorithm, identified the econnoisseur with similar characteristics into groups. When the econnoisseur characteristics change this method fails.

The challenges of e-commerce econnoisseur identification are as follows: 1. Rule omission. at present, most platforms intercept econnoisseurs based on rules, but the detection will be missed when econnoisseur behavior changes. 2. Insufficient sample labels. The econnoisseur is newly added every day, and manual labeling requires a large labor cost. 3. Model detection lag. With the iterative update of the econnoisseur's method, the existing rules and models have a certain lag, so it is necessary to periodically iterate and maintain the model.

3 Theoretical Basis

Anomaly detection can be divided into supervised and unsupervised anomaly detection. Since the econnoisseur is generated in real-time, unlabeled sample data is mainly used in practical applications, this paper uses the unsupervised anomaly detection scheme. Considering the different applicability of detection methods in different scenarios and data, this paper compares three commonly used anomaly detection schemes to see their ability to identify econnoisseur in e-commerce website log data.

3.1 Isolated Forest (IForest)

The isolation forest model was first proposed by Zhou Zhihua's team and Fei Tony Liu of Monash University [13] as an ensemble learning method. It was used in the field of industrial anomaly detection due to the advantages of high accuracy and linear time complexity. The theoretical basis of the model are: 1. There are differences between abnormal data and normal data. 2. The proportion of abnormal data is relatively small. These two theories are consistent with the econnoisseur detection. The Isolation Forest algorithm cuts the data space through a random hyperplane. The data plane can divide the data into two subspaces at a time, until each subspace has only one sample point or reaches the given height of the tree.

The Isolated Forest needs to be trained on the Isolated Tree first to obtain the Isolated Forest. After that, calculated the isolated score S of each test sample, then compare the difference between isolated score S and the given threshold to see whether the sample is an abnormal sample.

Algorithm 1. Isolated Forest training

Input: X -input data, t -number of trees, ψ -subsampling size

Output: a set of t iTrees

- 1: Initialize Forest
 - 2: set tree height limit $l = \text{ceiling}(\log_2 \psi)$
 - 3: **for** $j = 0$ to t **do**
 - 4: $X' = \text{sample}(X, \psi)$, randomly select subsample ψ
 - 5: Initialize $iTree_i$, tree depth $e = 0$
 - 6: **while** $|X'| > 1$ and $e < l$
 - 7: randomly select an attribute q from attributes Q
 - 8: randomly select a split point p from max and min values of attribute q in X
 - $X_l \leftarrow \text{filter}(X, q < p)$
 - $X_r \leftarrow \text{filter}(X, q \geq p)$
 - $\text{inNode}\{\text{Left} \leftarrow \text{filter}(X'_l, e + 1, l),$
 - $\text{right} \leftarrow \text{filter}(X'_r, e + 1, l),$
 - $\text{SplitAtt} \leftarrow q,$
 - $\text{SplitValue} \leftarrow p\}$
 - 10: **end while**
 - 11: $\text{Forest} \leftarrow \text{Forest} \cup iTree(X')$
 - 12: **end for**
 - 13: **return** Forest
-

After data training to obtain an Isolated Forest, the anomaly score of the test sample can be evaluated based on the generated Isolated Tree. Since the structure of the Isolation Tree is consistent with the binary search tree (BST), the average path length of the tree is consistent. Based on this, BST is used to estimate the average path length of isolated tree.

$$c(n) = 2H(n - 1) - (2(n - 1)/n) \quad (1)$$

$$H(n) = \ln(n) + 0.5772156649 \quad (2)$$

Formula (1) is the average path depth of the isolated tree composed of n samples, and it is used to standardize the depth of the samples on the Isolated Tree, so that the abnormal score of the test sample x is shown in Formula (3).

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (3)$$

$$E(h(x)) = \sum_{i=1}^t h_i(x)/t \quad (4)$$

Algorithm 2. Calculate the sample anomaly score

Input: x - an instance, Forest-Isolated Forest

Output: anomaly score s

1: Initialize tree depth $h(x) = []$

2: **for** $i = 0$ to t **do**

3: extract the i th Isolated Tree $iTree_i$, initialize tree height $e = 0$

4: **if** $iTree_i$ is an external node

5: $h_i(x) = e + c(iTree_i.size)$, $c(\cdot)$ is defined in Equation(1),

6: **end if**

7: $a \leftarrow iTree_i.splitAtt$

8: **if** $x_a < iTree_i.splitValue$

9: $h_i(x) = PathL(x, iTree_i.left)$

10: **else**

11: $h_i(x) = PathL(x, iTree_i.right)$

12: **end if**

13: **end for**

14: calculate anomaly score $s(x, n)$, $s(\cdot)$ is defined in Equation(3)

3.2 Local Outlier Factor (LOF)

The Local Outlier Factor [14] is a model that determines whether the sample points are abnormal based on the density. Its core concept is that the density of abnormal points is smaller than that of other points. Before introducing the Local Outlier Factor model, we need to understand some basic concepts.

Definition 1: (*k*-distance). For point p , sort the distances between point p and other points from small to large, and the k -th closest distance point to point p is *k*-distance of point p . If point o is the k -th point closest to point p , then distance is *k*-distance of object p , i.e.

$$k_distance(p) = d(p, o) \quad (5)$$

Definition 2: (*k*-distance neighborhood). Draw a circle with point p as the center and *k*-distance as the radius. The points in this circle is the *k*-distance neighborhood of p , i.e.

$$N_k(p) = \{d(p, o') \leq d_k(p)\} \quad (6)$$

Definition 3: (reachability distance). Take point o as the center, and take the maximum value of the k -th distance nearest to point o , then the distance is the reachable distance from point p to point o .

$$reach_dist_k(o, p) = \max\{d_k(o), d(o, p)\} \quad (7)$$

Definition 4: (local reachability density). The reciprocal of the average reachable distance in the neighborhood of point p is the local reachable density of point p , defined as

$$lrd_k(p) = \frac{1}{\frac{\sum_{o \in N_k(p)} reach_dist_k(p, o)}{|N_k(p)|}} \quad (8)$$

Definition 5: (local outlier factor). The mean of the local reachability density of points in the field divided by the local reachability density of point p is the local outlier factor of point p , defined as

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd(o)}{lrd(p)}}{|N_k(p)|} \quad (9)$$

Algorithm steps:

1. Calculate the distance between each sample point and all other points, sort them from near to far.
2. For each sample point, find the point in its *k*-distance field, and then calculate its LOF score.
3. Given the threshold, if the LOF value of the sample point is higher than the threshold, the sample point is an abnormal point.

Therefore, the algorithm calculates the density of the samples based on the local points in the k field of the sample points. The lower the density, the greater the probability of abnormal samples.

3.3 DBSCAN

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) [15] is a density based spatial clustering algorithm.

The concepts involved in the algorithm are:

Definition 1: (Core user). For sample point p , give a distance ε , if there are at least $Minpts$ sample points within ε neighborhood, then p is the core point. For point p , its density is defined as $\rho(p) = |N_\varepsilon(p)|$. Where $N_\varepsilon(\cdot)$ denotes the set of points in its ε neighborhood. If p is a core user, then defined it as.

$$\rho(p) = |N_\varepsilon(p)| \geq Minpts \quad (10)$$

Definition 2: (Directly density-reachable). If point p and point q is directly density-reachable, the following two conditions must be satisfied:

i) p is in the ε neighborhood of a core point $q \in N_\varepsilon(q)$. . ii) q is core user $|N_\varepsilon(q)| \geq Minpts$.

Definition 3: (Density-reachable). If there is a point o so that both p and q can be directly density-reachable, then the point p and q densities is density-reachable.

Definition 4: (Border user). For sample point p , if the sample points included in the ε radius are smaller than $Minpts$ and the sample is in the field of other core points, sample p is the border user.

Definition 5: (Noise user). If sample point p is of non core user and border user, then it is names noise user.

Algorithm 3. DBSCAN

Input: P -sample data, ε -scan radius, $Minpts$ -minimum number of points included
Output: sample category set $X = \{x_1, x_2, \dots, x_n\}$

- 1: set sample category $x_i = 0, 1 \leq i \leq n$
number of categories $k=1, h = \varnothing$
- 2: **while** $P \neq \varnothing$
- 3: randomly take a user p_i from P
- 4: **if** $x_i = 0$
- 5: **if** $\rho(p_i) = |N_\varepsilon(p_i)| < Minpts$
- 6: $x_i = -1$
- 7: **else**
- 8: $x_i = k$
add users in $N_\varepsilon(p)$ into h ;
- 9: **while** $h \neq \varnothing$
- 10: randomly select h from a user P_j
- 11: **if** $x_j = 0$ or -1
- 12: $x_j = k$
- 13: **end if**
- 14: **if** $\rho(p_j) = |N_\varepsilon(p_j)| \geq Minpts$
- 15: add users in $N_\varepsilon(p)$ into h ;
- 16: **end if**
- 17: **end while**
- 18: $k = k + 1$
- 19: **end if**
- 20: **end if**
- 21: **end while**

4 Research Process

4.1 Data Sources

The data is based on the real-time streaming log data in the Knownsec Security Intelligence Brain. The fields in the log data are: access time, access IP, user agent, URL link, website domain name, etc. Three e-commerce websites log data was screened, an IP and user agent was regarded as an independent visitor of the website.

Log data with abnormal access URLs or few visits is deleted to reduce interference to the model. The data of three e-commerce websites in November 2021 are sampled for observation to see their performance in one month. The website visit situation is shown in Table 1 :

Table 1. Website visit number.

Website	Daily average visits	Daily average visited users
Web A	6,621,059	155,070
Web B	1,298,633,918	29,644,844
Web C	51,625,887	2,539,750

4.2 Feature Construction

After analyzing the access data of some users, it is found that the econnoisseur can be divided into two categories: 1. Monitoring users, who monitor the preferential information of commodities in real-time and at low frequency. 2. Activity type users, who make high-frequency application and purchase of goods with large discounts during the activity period.

Combing with the characteristics of the econnoisseur, we constructed three characteristics for users: website visits, website visit time and different website visits, totally nine features are shown in Table 2.

Table 2. User features

Feature category	Feature name	Explain
Website visit	Visit PV	Total visit number
	URL visits	Remove duplicate URL visits number
	URL concentration	TOP three URL visit number divided by total visit number
Website visit time	Total visit time	Time interval between start and end time
	Visit periods	10 min as a time period, if the interval between adjacent access points is more than 10 min, time period is increased
	Average visit time	Total visit time divided by visit periods
Different website visits	current website visit ratio	Current website visit number divided by user total visit number
	E-commerce websites ratio	E-commerce websites visit number divided by user total visit number
	E-commerce host ratio	E-commerce websites host visit number divided by total host visit number

4.3 User Behavior

Reduce the user feature data of website C to 2D for visualization based on t-SNE (t-distributed Stochastic Neighbor Embedding), as shown in Fig. 2. User data can be divided into four categories according to the color of points. Some sample data were extracted from the four categories and analyzed. It was found that the econnoisseur appeared in categories two and four, while the normal users were concentrated in categories one and three (Fig. 3).

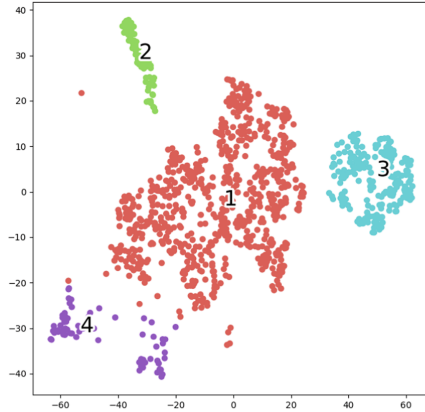


Fig. 2. User dimension reduction visualization

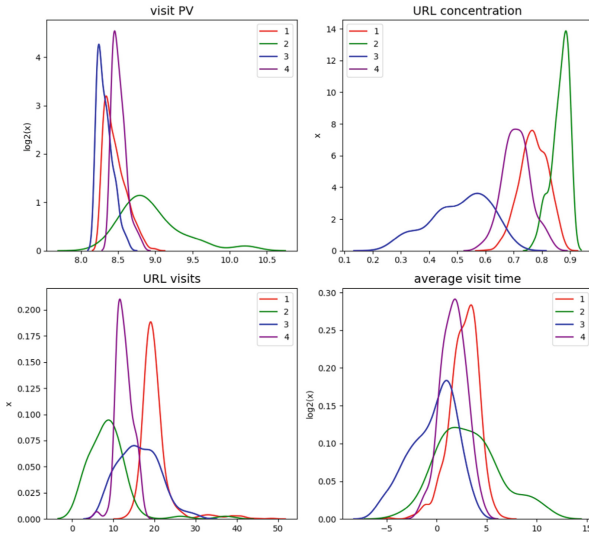


Fig. 3. Feature density curves of different categories

Draw a feature density map for the 4 categories of sample users, shown in Fig. 2. It can be seen that there are differences between the characteristics of different categories of users, that is.

Category 1: random visit users. Characterized by small visit number, short visit time, and less website visit information.

Category 2: monitoring users. Users visit specific web pages for a long time and infrequently.

Category 3: normal access users. The categories of web pages visited by users are scattered, and the time of visiting web pages is more than that of category 2, which is a normal browsing user of web information.

Category 4: specific page access users. A large number of visits to the website in a short period of time, and the visited pages are targeted.

4.4 Result Analysis

The data of the previous week of the website are used as the training set for model training, and the econnoisseur detection is carried out on the user access of the latest day. The daily update and retraining of training data can obtain the recent overall distribution of users, and the model can be adjusted in real-time. During the Double Eleven period, these e-commerce companies had promotional activities, which also became the carnival of the econnoisseur. The user access data on November 11 and November 18 were extracted to compare the detection effect of econnoisseur during the active period and the non-active period.

Since the detected user data is unlabeled data and the sample size is large, 3000 users data are randomly selected from each website for labeling to check the test effect of the model. In Table 3, it can be seen that the detection precision and recall of the Isolated Forest model are higher than those of the other two models, showing that the model has better applicability.

After further analysis, it was found that some of the underreported econnoisseur were due to small difference between the amount of website URLs visited by econnoisseur and normal users. For example, aa/01 and aa/02, these two URLs are the same type of URLs, which can be combined to better to distinguish different user access situations. Consider combining URL of the same type based on the Bidirectional Long Short Term Memory (BiLSTM) model to reduce its impact on URL visits distribution. The detection effect of econnoisseur is shown in Table 4.

The average detected amount of econnoisseur during the period from November 1 to November 11 was taken as the average of daily econnoisseur amount during the activity period. The average detected amount of econnoisseur during the period from November 12 to November 30 was taken as the average of daily econnoisseur amount during the inactive period. The performance is shown in Fig. 4, it can be seen that the econnoisseur during the activity period is about twice as much as the non activity period.

It can be concluded from the above:

1. The Isolated Forest model performs better in precision and recall than the other two models in different e-commerce websites and active and inactive periods, it gave a good result in econnoisseur detection.
2. After combining URL of the same type with the BiLSTM model, the detection ability of the BiLSTM-IForest model to the econnoisseur is significantly higher than that of the Isolated Forest in the recall rate.

The detection ability of econnoisseur during non-activity period is better than that during activity period. According to the analysis, it is found that some users have visited the preferential products for many times with low frequency. This behavior is similar to that of normal users and has not been detected. It can be further optimized later.

Table 3. Model result comparison

Activity	Model	Precision			Recall			F1-Score		
		Web A	Web B	Web C	Web A	Web B	Web C	Web A	Web B	Web C
Activity period	IForest	0.74	0.79	0.91	0.72	0.75	0.82	0.73	0.77	0.86
	LOF	0.68	0.75	0.84	0.57	0.74	0.7	0.62	0.74	0.76
	DBSCAN	0.6	0.56	0.68	0.62	0.6	0.66	0.61	0.58	0.67
Non-activity period	IForest	0.80	0.83	0.92	0.74	0.81	0.83	0.77	0.82	0.87
	LOF	0.72	0.77	0.85	0.6	0.78	0.74	0.65	0.77	0.79
	DBSCAN	0.65	0.57	0.66	0.66	0.61	0.67	0.65	0.59	0.66

Table 4. Comparison of results before and after improvement of Isolated Forest

Activity	Model	Precision			Recall			F1-Score		
		Web A	Web B	Web C	Web A	Web B	Web C	Web A	Web B	Web C
Activity period	IForest	0.74	0.79	0.91	0.72	0.75	0.82	0.73	0.77	0.86
	BiLSTM-IForest	0.75	0.82	0.92	0.77	0.81	0.87	0.76	0.81	0.89
Non-activity period	IForest	0.80	0.83	0.92	0.74	0.81	0.83	0.77	0.82	0.87
	BiLSTM-IForest	0.81	0.85	0.94	0.8	0.83	0.89	0.80	0.84	0.91

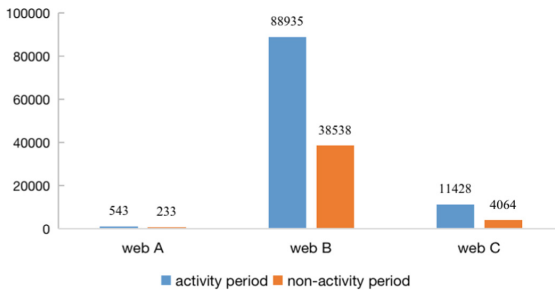


Fig. 4. Detection amount of econnoisseur users in different periods

5 Conclusion

The rise of e-commerce platforms not only brings convenience to people's life, but also poses a higher challenge to the platform's risk control ability. How to control and reduce the risk of the platform being played for a sucker needs to be solved urgently. Based on the real log data of website users visiting the website in the Knownsec Security Intelligence Brain, this paper extracts nine features of users to identify the econnoisseur.

By comparison, it is found that the unsupervised anomaly detection model has certain detection ability for the e-commerce website econnoisseur. Among them, Isolated Forest has higher detection precision and recall rate than LOF and DBSCAN models in three e-commerce websites, and is more suitable for current e-commerce user data.

After that, the analysis found the same type of URL visited by users, which can be combined to better describe the real visit behavior of users. The detection results show that the econnoisseur detection based on the BiLSTM-IForest has been further improved.

The econnoisseur selected in this paper can intercept its traffic access in advance. After that, a risk control model can be built based on the actual browsing, purchasing and other specific behaviors of users to strengthen real-time prevention and control.

The econnoisseur and the platform have always been in a state of mutual competition. The so-called the devil is one foot tall and the road is one foot tall. We need to track the attack methods of the econnoisseur in real-time, and at the same time combine the risk control platform to further attack the econnoisseur.

References

1. Bohara, A., Noureddine, M.A., Fawaz, A., et al.: An unsupervised multi-detector approach for identifying malicious lateral movement. In: *Reliable Distributed Systems*. IEEE (2017)
2. Yao, Z.: Research and implementation of advanced persistent threats detection method based on data mining. Beijing University of Posts and Telecommunications (2019)
3. Ren, X., Jiao, W., Zhou, D.: Intrusion detection model of Weighted Navie Bayes based on Particle Swarm Optimization algorithm. *Comput. Eng. Appl.* **52**(7), 122–126 (2016)
4. Sun, W., Zhang, P., He, Y., et al.: Attack detection method based on spatiotemporal event correlation in intranet environment. *J. Commun.* **41**(01), 33–41 (2020)
5. Duan, X., Jia, C., Liu, C.: Intrusion detection method based on hierarchical hidden Markov model and variable-length semantic pattern. *Journal on Communications* **31**(03), 109–114 (2010)
6. Li, X., Zhu, M., Yang, L.T., et al.: Sustainable ensemble learning driving intrusion detection model. *IEEE Trans. Dependable Secure Comput.* **PP**(99), 1–1 (2021)
7. Otair, M., Ibrahim, O.T., Abualigah, L., et al.: An enhanced Grey Wolf Optimizer based Particle Swarm Optimizer for intrusion detection system in wireless sensor networks. *Wireless Netw.* **28**, 721–744 (2022)
8. Yang, J.: An improved intrusion detection algorithm based on DBSCAN. *Microcomput. Inf.* **25** (2009)
9. Shamshirband, S., Amini, A., Anuar, N.B., et al.: D-FICCA: a density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *Measurement* **55**, 212–226 (2014)
10. Veeraiyah, N., Krishna, B.T.: Trust-aware FuzzyClus-Fuzzy NB: intrusion detection scheme based on fuzzy clustering and Bayesian rule. *Wireless Netw.* **25**(8), 1–15 (2019)

11. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. *VLDB J.* **16**(4), 507–521 (2007)
12. Dandan, Y.: Research and application of community detection algorithm based on node importance. Xidian University(2020)
13. Fei, T.L., Kai, M.T., Zhou, Z.H.: Isolation Forest. In: *IEEE International Conference on Data Mining*. IEEE (2008)
14. Breunig, M.M., Kriegel, H.P., Ng, R.T., et al.: LOF: identifying density-based local outliers. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2000)
15. Bäcklund, H., Hedblom, A., Neijman, N.: A density-based spatial clustering of application with noise. *Data Min.* (2011)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





CMPD: Context-Based Malicious Parameter Detection for APIs

Zhangjie Zhao¹, Lin Zhang¹, Xing Zhang², Ying Wang^{2(✉)}, and Yi Qin²

¹ Beijing Big Data Centre, Beijing 100101, China
{zhaozj,zhanglin}@jxj.beijing.gov.cn

² National Engineering Laboratory for Big Data Collaborative Security Technology,
Beijing 102209, China
{zhangxing,wangying,qinyi}@cecgw.cn

Abstract. The Application Program Interface (API) plays an important role as the channel for data interaction between programs, while the widespread use of APIs has brought security risks that cannot be ignored. The adversary can perform various Web attacks, including SQL Injection and Cross-Site Scripting (XSS), by tampering with the parameters of API. Efficient detection of parameter tampering attacks for API is critical to ensure the system is running in the expected condition, further avoiding data leakage and property loss. Previous works always utilize the rule-based method or simple learning-based method to detect parameter tampering attacks. However, they ignore the contextual information of the API tokens and thus have a poor performance. In this paper, we propose the **C**ontext-based **M**alicious **P**arameter **D**etection (CMPD) framework to detect the parameter tampering attacks for APIs. We use a neural network language model to learn the distribution of the parameters, parameter names, and URLs and then use a tree model to detect the malicious query based on the high dimensional API embedding. Experiments show that CMPD outperforms all baseline, including rule-based method, Support Vector Machine (SVM), and Autoencoder, on CSIC 2010 dataset with F_1 value reaching 0.971. CMPD can also achieve a 0.895 F_1 value when training data is reduced to 20% and can achieve a 0.910 F_1 value when negative examples are reduced to 1%.

Keywords: Parameter tampering · API · Language model

1 Introduction

The API is a combination of a set of definitions and protocols, which plays an important role as a channel for data interaction between programs. Modern applications are often developed with many well-defined interfaces to improve the scalability and compatibility of the program. Although the widespread use of APIs has brought great convenience to data access, and thus different terminals can access relevant information in a similar way, the extensive use of APIs has also brought security issues that cannot be ignored. Especially in the modern

microservice architecture, each application is subdivided as much as possible, making the security risks faced by APIs more difficult to be detected completely. Effective detection of API security risks ensures that the system is running in good condition.

The access of API is based on the HTTP/HTTPS protocol, so the threat on the Web protocol may extend to API, such as SQL Injection, Broken Authentication, Session Management, Cross-Site Scripting (XSS). It should be noted that these attacks are always implemented by tampering with the parameters in the API. Therefore, to mitigate the threat of API, a key idea is to prevent the parameters from being tampered with. The security community has proposed a variety of approaches to address the security risks of parameter tampering, the most common of which is the rule-based detection. Such methods are often implemented by a lightweight agent that first detects security risks that may be contained in a Web request before a server process it. If a relevant rule is matched and a request is identified as a security risk, the request is filtered out to avoid the server from being affected. Although this detection method is simple to implement and efficient, its over-reliance on rules that humans preset leads to its inability to detect unknown new attacks, and thus not only has a poor performance in actual detection but also has a high false-negative rate.

Deep learning model has achieved remarkable success in various natural language processing (NLP) tasks, and it has been shown that these models can effectively learn the data distribution, which is difficult for the rule-based detection method to do. By learning the data distribution, the detection of parameter tampering can thus be seen as a pattern classification problem whose goal is to distinguish the feature pattern of the normal access of API and the malicious access of API. However, deep learning methods always need to learn from a large amount of data, which may be difficult to obtain. Furthermore, normal access is more common than malicious access, and the ratio of normal accesses and malicious accesses may be significant unbalance. The unbalance data also makes the model difficult to learn the data distribution, as the model may mainly focus on the type of data that is more common in the dataset and ignores the less one.

In this paper, to detect the parameter tampering attack against API and reduce the influence of unbalanced data, we propose the Context-based Malicious Parameter Detection (CMPD) framework. CMPD improves the effectiveness of detecting malicious parameters by learning the distribution of each component of the API and builds the relationship amount URL, parameter names, and parameters. Experiments show that CMPD outperforms all baseline on CSIC 2010 dataset, with F_1 value reaching 0.97. CMPD also achieves 0.91 F_1 value on the unbalance dataset that normal access data is 100 times more than the malicious data, and achieves 0.89 F_1 value when training data of CSIC 2010 dataset are reduced to 20%.

We summarize our main contributions as follows:

- We propose a semantic extraction and learning module to learn the relationship amount URL, parameter names, and parameters, which universally models the parameter distribution of different APIs in one framework.

- We propose the Context-based Malicious Parameter Detection (CMPD) framework, which can effectively detect parameter tampering attacks against API based on the context information indicated by the distribution of the parameter.
- Experiments on the CSIC 2010 dataset show that CMPD outperforms other baselines, including rule-based methods, Support Vector Machine (SVM), and Autoencoder, and achieves competitive results on unbalanced data and reduced data.

2 Related Work

2.1 Vulnerability Detection for APIs

To detect API’s vulnerability, current methods mainly focus on black-box testing, which needs to generate a large number of testing cases. Much research relies on crawlers or manual methods to get the detection object, parse out the fuzz domain based on the detection object to generate test cases, and use the attack pattern library to perform vulnerability detection [1, 3, 4]. To generate test cases, various methods are proposed. Atlidakis et al. [2] propose REST-ler to automatically generate test requests with a random walk algorithm. Avinash [12] et al. proposed six attack patterns for replay attacks to automatically generate test cases. Douibi et al. [5] automatically generate test cases for REST API based on the description of Swagger and OpenAPI. Because the crawler-based API vulnerability detection method has the problem of low coverage and manual testing can not be carried out on a large scale, the black box testing method is often combined with the interface documentation. Yu et al. [16] propose a fuzz system with RESTful API based on SwaggerHub’s development interface and improves the effectiveness of fuzz testing by automatically generating test cases and automatic filtering. Viglianisi et al. [14] generated normal test cases and malicious test cases based on the interface documentation to test the security risk of RESTful API. Different tools have also been proposed to automatically scan API vulnerabilities, such as FuzzAPI¹, APIFuzzer², boofuzz³, and Astra⁴. These tools do not need to obtain source code and interface documents but combine manual and crawler methods to achieve vulnerability detection. When interface documents are available, the tools such as TNT-Fuzzer⁵, 42Crunch⁶, and OWASPZAP⁷ can directly extract detection objects from interface documents to achieve vulnerability detection with high coverage.

¹ <https://github.com/Fuzzapi/fuzzapi>.

² <https://github.com/KissPeter/APIFuzzer>.

³ <https://github.com/jtpereyda/boofuzz>.

⁴ <https://github.com/flipkart-incubator/Astra>.

⁵ <https://github.com/Teebytes/TnT-Fuzzer>.

⁶ <https://42crunch.com>.

⁷ <https://www.zaproxy.org>.

2.2 Parameter Tampering Detection for APIs

To detect the parameter tampering attacks for APIs, both rule-based methods and learning-based methods are proposed. ModSecurity⁸ develops the OWASP ModSecurity Core Rule Set (CRS), which contains a large number of rules for detecting SQL Injection, Cross-Site Scripting, and HTTP Protocol Violations. Rieck et al. [11] use the n-grams and a similarity measurement to generate new features for anomaly detection. Ingham et al. [6] proposed the Deterministic Finite Automata (DFA) induction method, which uses a heuristic algorithm to detect abnormalities. Ma et al. [8] use machine learning methods including Naive Bayes, Support Vector Machine, and Logistic Regression to learn the distribution of static features to detect attacks. Nguyen et al. [10] use a feature selection algorithm to reduce the dimension of features extracted from traffic, reducing the computational complexity of the learning algorithm. Liang et al. [7] developed an RNN-MLP network to detect malicious accesses, where the RNN contains LSTM and GRU cells, and the MLP follows the RNN. Wang et al. [15] investigated CNN and LSTM and their combination method for malicious detection, which outperforms the traditional methods.

3 Methodology

3.1 Parameter Tampering Attacks Against APIs

API parameter attacks attempt to manipulate parameters transmitted between the client and server in order to alter application data, such as user passwords and permissions, product prices and quantities. This type of data is typically kept in cookies, hidden form fields, or URL query strings and is used to regulate and enhance the functionality of the program. The attack’s success is conditional on integrity and logical validation mechanism faults, and exploiting these errors may result in further implications such as cross-site scripting (XSS) and SQL injection. The tampering of parameters is frequently limited to several essential categories of data: API query parameters, cookies, form fields, and HTTP headers. Specifically, for an API, which is consisted of a basic URL u , a group of parameter names $\{n_i | i \in N\}$, and a group of parameter $\{p_i | i \in N\}$. The i -th parameter is integrated with the i -th parameter name. Suppose the server expects to receive a benign query, and for the target u , all possible benign choices of the i -th parameter are denoted as \mathcal{P}_i , all possible benign choices of the i -th parameter name are denoted as \mathcal{N}_i . Therefore, a benign API query for the target URL u can be defined as

$$\forall i \in N, p_i \in \mathcal{P}_i, \quad \text{and} \quad \forall i \in N, n_i \in \mathcal{N}_i \quad (1)$$

And a parameter tampering attack for the target URL u can thus be defined as

$$\exists i \in N, p_i \notin \mathcal{P}_i, \quad \text{and} \quad \exists i \in N, n_i \notin \mathcal{N}_i \quad (2)$$

⁸ <https://github.com/SpiderLabs/ModSecurity>.

Intuitively, a parameter tampering attack may happen in the following conditions:

- The adversary tampers the parameters of an API, attempting to make the server process the tampered parameters for malicious purposes such as SQL injection and XSS attacks.
- The adversary tampers the parameters name of an API, attempting to let the server process the tampered parameter names to achieve malicious purposes such as bypassing verification.
- The adversary tampers both the parameters and the parameters name of an API. Even if the tampered parameters and parameter names are benign values for another API, they are indeed malicious values for the current API.

3.2 Semantic Extraction and Learning

As parameter tampering attacks have the characteristics of a wide attack surface and large scope of tampering, traditional methods cannot realize the judgment of whether a request has been tampered with in one model. Furthermore, as text information is discrete, traditional methods cannot use the semantic information contained in it. Therefore, we use the Semantic Extraction and Learning Module to learn the distribution relationship among the basic URL in the API, the parameter names, and the parameters and then map discrete text information to high-dimensional continuous space. The general frameworks of the semantic extraction and learning module is shown in Fig. 1.

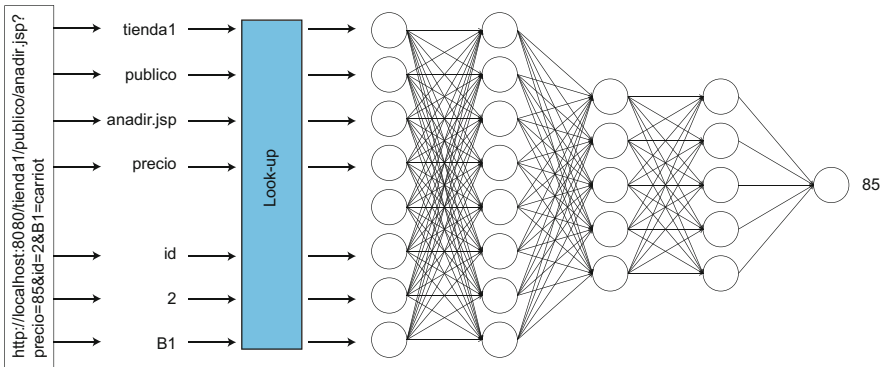


Fig. 1. Illustration of the semantic extraction and learning module of CMPD.

Specifically, suppose there is a neural network with K layers, and the weights and the bias vectors for each layer can be defined as

$$\begin{aligned}
 \mathbf{W}^{(1)} &\in R^{m_1 \times m_0} & \mathbf{b}^{(1)} &\in R^{m_1 \times 1} \\
 \mathbf{W}^{(2)} &\in R^{m_2 \times m_1} & \mathbf{b}^{(2)} &\in R^{m_2 \times 1} \\
 &\dots & & \\
 \mathbf{W}^{(K)} &\in R^{m_K \times m_{K-1}} & \mathbf{b}^{(K)} &\in R^{m_K \times 1}
 \end{aligned} \tag{3}$$

where (m_0, m_1, \dots, m_K) is the number of units in each layer. The active function in each layer are denoted as $(f^{(1)}, f^{(2)}, \dots, f^{(K)})$, and thus the output of the K -th layer $\mathbf{Y}^{(k)}$ can be defined as

$$\begin{aligned}
 \mathbf{net}_i^{(k)} &= \sum_{j=1}^{m_{k-1}} W_{i,j}^{(k)} Y_j^{(k-1)} + b_i^{(k)}, (1 \leq i \leq m_k) \\
 \mathbf{net}^{(k)} &= \mathbf{W}^{(k)} \mathbf{Y}^{(k-1)} + \mathbf{b}^{(k)} \\
 \mathbf{net}^{(k)} &= [\mathbf{net}_1^{(k)}, \mathbf{net}_2^{(k)}, \dots, \mathbf{net}_{m_k}^{(k)}]^T \\
 \mathbf{Y}^{(k)} &= f^{(k)}(\mathbf{net}^{(k)}) = [Y_1^{(k)}, Y_2^{(k)}, \dots, Y_{m_k}^{(k)}]^T
 \end{aligned} \tag{4}$$

We extract the URL u , the parameter names $\{n_i | i \in N\}$, and the parameter $\{p_i | i \in N\}$ in each API query and then arrange them in the order they appear in the query as $(w_t)_{t \in \{1, 2, \dots, M\}}$. We randomly remove a token w_t in $(w_t)_{t \in \{1, 2, \dots, M\}}$, and send the rest token into the network with a look-up layer that is concatenated before the first layer. The look-up layer has the parameter in dimension $V \times E$, where V is the size of vocabulary, and E is the size of embedding. This module maps the token to continuous values. We expect the network knows what the removed token is and output the probability of the removed word in $\mathbf{Y}^{(k)}$. Turning the training, the probability of the removed word in the output layer, i.e., the K -th layer, is maximized. After training, we use the value in the look-up layer as the embedding of a token and the average result of each token in an API query as the embedding of the API. In this high-dimensional continuous space, the representation of tokens indicates the relationship between tokens so that subsequent modules can effectively use the semantic information in the token.

3.3 Detection on Parameter Tampering

To reduce the reliance of model on the amount of data and to enable it to learn effectively when the positive and negative samples are not balanced, we additionally classify the API embedding using a decision tree model.

A decision tree model is a tree structure that describes how instances are classified, and it is composed of nodes and directed edges. Nodes are classified into two types: internal nodes and leaf nodes. Internal nodes denote a property

or attribute, while leaf nodes denote a class. Begin with the root node and test a specific feature of the instance; then, using decision tree classification, assign the instance to its child nodes based on the test results; at this point, each child node corresponds to a value of the feature. Recursively, instances are tested and allocated until a leaf node is reached. Specifically, suppose the training data consisting of all the API embedding is D , A is the feature group, C_k is the samples of class k , the dataset can thus be separated into D_1, D_2, \dots, D_n . Denote the samples in D_i and in class C_k as D_{ik} , the entropy of the dataset D can be calculated as

$$H(D) = - \sum_{k=1}^K \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|} \quad (5)$$

and the conditional entropy of D given A is

$$H(D | A) = \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i) = - \sum_{i=1}^n \frac{|D_i|}{|D|} \sum_{k=1}^K \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|} \quad (6)$$

and the information gain of D from A is defined as

$$g(D, A) = H(D) - H(D | A) \quad (7)$$

During the training, the attribute with the largest information gain rate is selected as the test attribute each time, and the construction of the decision tree is completed from top to bottom. The Parameter Tampering Detection Module in CMPD is consisted of the well-learned decision tree.

3.4 Context-Based Malicious Parameter Detection Framework

Based on the previous analysis, we now illustrate the general architecture of the proposed Context-based Malicious Parameter Detection (CMPD) framework, which is shown in Fig. 2.

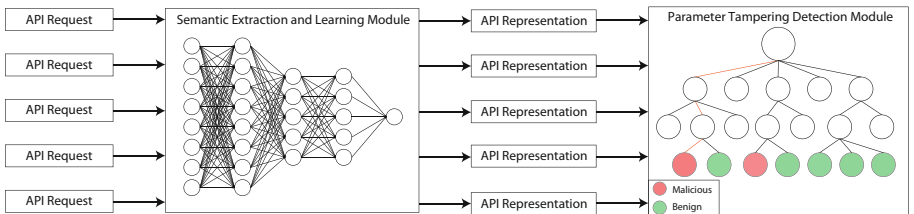


Fig. 2. General architecture of CMPD.

The CMPD framework is consisted of the semantic extraction and learning module we detailed in Sect. 3.2, and the parameter tampering detection Module we detailed in Sect. 3.3. We first collect all the API access records in the form of

URL requests and feed all of the collected data into the semantic extraction and learning module. Therefore, the API request will be map to the high dimensional hidden space in the form of API vector representation, which contains the context information about the normal and abnormal parameters. We then collect all the API representations and feed them to the parameter tampering detection module, which can complete the malicious parameter detection without relying on the balanced and numerous data. The detection module will classify each API representation into a benign request or an abnormal request from the root to leaf nodes of the decision tree, as we illustrated in Fig. 2.

4 Experiments

4.1 Metric

The experiments will focus on the identification of the parameter-tampering attacks, and the evaluation metrics used in the current work include precision, recall, F_1 . These metrics are calculated using the proportion of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in the classification results. TP and TN are the number of correctly classified malicious and legitimate API requests. FP is the numbers of normal API requests misclassified as malicious, while FN is the number of abnormal requests misclassified as legitimate API requests. Where the precision is calculated as

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

the recall is calculated as

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

the F_1 value is calculated as

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

4.2 Main Result

The results of different methods on the HTTP DATASET CSIC 2010 are shown in Table 1. CRS stands for Core Rule Set, and PL stands for Paranoia Level, which is used to control the strictness of ModSecurity’s rule checking, with a smaller value indicating greater strictness. As the PL increases, the Precision value of ModSecurity increases while the Recall value decreases, resulting in a decrease in the F_1 value, indicating that the traditional method based on rules has a very limited effect. The effect of the SVM algorithm on detecting parameter tampering is weaker than that of the traditional method, most likely because the SVM algorithm is extremely dependent on the quality of the features, and the

features fail to indicate the distribution of data. Autoencoder are deep learning-based methods that perform better than traditional methods. The proposed CMPD outperforms all baselines, including traditional detection methods and learning-based methods, in terms of F_1 . CMPD has a balanced precision and recall, indicating that our method has low false-positive and false-negative rates.

Table 1. The comparisons on Precision, Recall, and F_1 Score.

Method	Precision	Recall	F1
Modsecurity + CRS (PL = 1)	1	0.652	0.789
Modsecurity + CRS (PL = 2)	0.936	0.685	0.792
Modsecurity + CRS (PL = 3)	0.841	0.745	0.79
Modsecurity + CRS (PL = 4)	0.682	0.791	0.732
One-class SVM + 30 features [10]	0.596	0.587	0.592
Stacked Autoencoder+Isolation Forest [13]	0.803	0.883	0.841
Regularized Deep Autoencoder [9]	0.946	0.946	0.946
CMPD	0.971	0.971	0.971

4.3 Further Analysis

Influence of the Number of Training Data. In practical situations, the samples of normal and abnormal accesses may be extremely unbalanced. To explore the effect of our model in a more demanding environment, we conducted experiments in two ways. We first reduce the number of training data to illustrate the performance of CMPD when training data is not enough. The influence of the number of training data is shown in Fig. 3. We find that the classification performance of the model gradually increases with the increase of training data, and the classification results are consistent with the results in Table 1. When the complete training data set is used, CMPD achieves the best classification performance. Moreover, when the training data is only 20% of the original dataset, the F_1 value can also achieve 0.89, indicating that CMPD is less sensitive to the amount of training data and is effective even with fewer data.

Influence of the Ration Between the Number of Negative Examples and Positive Examples. Further, we randomly drop the samples of malicious queries in the dataset, and the performance of our method are shown in Fig. 4. We find that when the number of malicious samples decreases, the performance of the classification decreases accordingly, but even when it is reduced to 1% of the normal samples, the F_1 value can still reach above 0.91, indicating that our model is effective even when the normal and malicious samples are extreme imbalance.

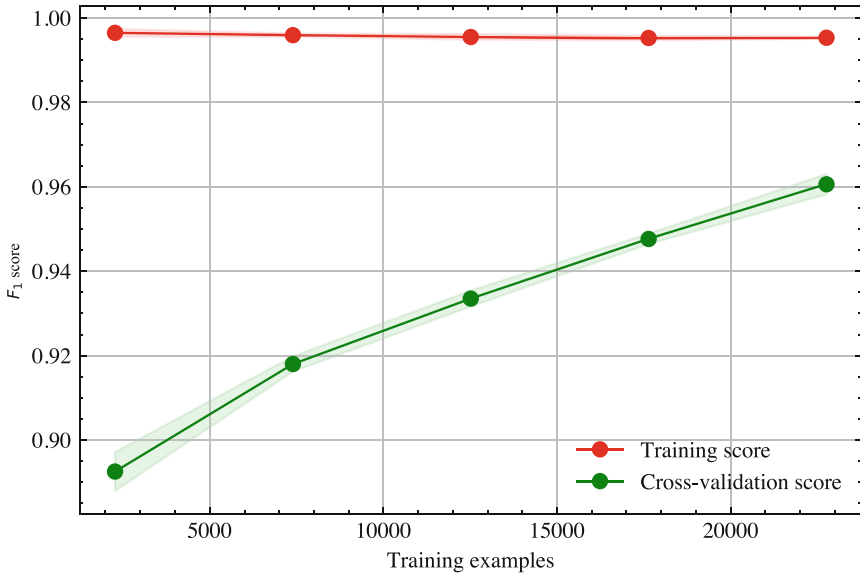


Fig. 3. Influence of the percentage of training examples.

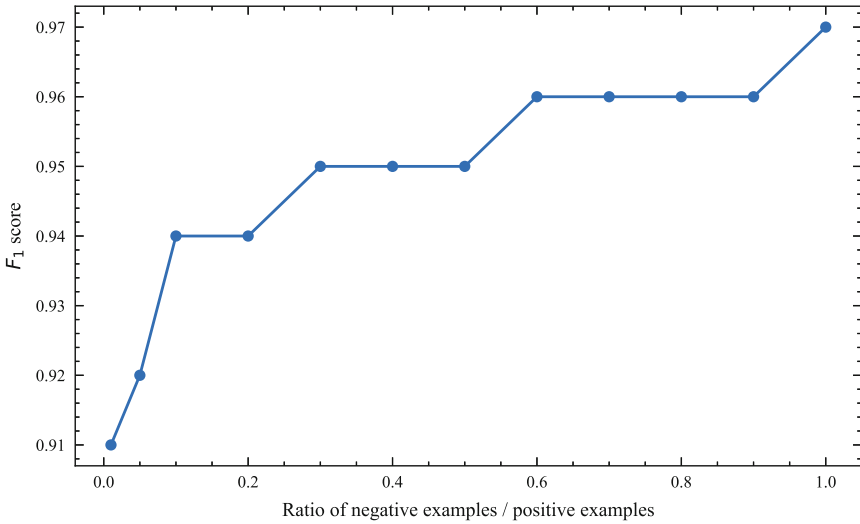


Fig. 4. Influence of the ration of negative examples/positive example.



Fig. 5. Visualization of the parameters and parameter names that the model most concentrated on.

Visualization of Model Concentration. Further, we collect the parameters and the parameter names that have a critical impact on the parameter tampering detection extracted by the model, and the visualization of these tokens is shown in Fig. 5. It can be seen that parameters such as *email*, *login*, and *password*, which are highly relevant to parameter tampering attacks, are correctly extracted and are considered to be of high importance, indicating that the model has different levels of attention to different parameters and successfully learns the features related to parameter tampering.

Case Study. To show that our method can identify the tampering with parameters, the tampering with parameter names, and the correspondence of URL and parameters or parameter names, we provide the results of case study in Table 2. As we illustrated in the Table 2, if we tamper with the parameter of the API “<http://localhost:8080/tienda1/publico/entrar.jsp>” and add the string “%11” behind the normal parameter “*errorMsg=Credenciales+incorrectas*”, the CMPD framework will find that the parameters of the API are tampered. Similarly, if we tamper with the parameter name (from *errorMsg* to *errorMsgBAC*), CMPD still successfully detects the tampering, which shows that our method learns the correct relationship between the parameters and the parameter names. Furthermore, if we change the parameter of the “<http://localhost:8080/tienda1/publico/entrar.jsp>” to the normal parameter of

Table 2. Case study on different types of tampering

URL	Parameters & Parameter names	Tampering type	Result
http://localhost:8080/tienda1/publico/entrar.jsp	errorMsg=Credenciales+incorrectas	/	Benign
http://localhost:8080/tienda1/publico/entrar.jsp	errorMsg=Credenciales+incorrectas%11	Tampering on parameter	Malicious
http://localhost:8080/tienda1/publico/entrar.jsp	errorMsgBAC=Credenciales+incorrectas	Tampering on parameter name	Malicious
http://localhost:8080/tienda1/publico/vaciar.jsp	B2=Vaciar+carrito	/	Benign
http://localhost:8080/tienda1/publico/entrar.jsp	B2=Vaciar+carrito	Parameter and parameter names does not correspond to the URL	Malicious

another API, “http://localhost:8080/tienda1/publico/vaciar.jsp”, our method can also detect that the parameter and parameter names do not correspond to the correct URL. It is shown that CMPD successfully learns the correspondence between URLs, parameters, and parameter names, and we do not need to use different models to detect the parameter tampering attack for different APIs.

5 Conclusion

APIs are vital for data exchange between programs, but their widespread use has brought significant security risks. By modifying API parameters, the adversary can launch Web attacks such as SQL Injection and Cross-Site Scripting (XSS). API parameter tampering detection is critical to keep the system running smoothly. To detect parameter tampering attacks, previous works always used rule-based or simple learning-based methods, while they ignore the API tokens’ contextual information and thus perform poorly. In this paper, we propose a framework for detecting API parameter tampering attacks called Context-based Malicious Parameter Detection (CMPD). We first learn the distribution of parameters, parameter names, and URLs using a neural network language model and then use a tree model to detect malicious queries based on the high-dimensional API embedding. On the CSIC 2010 dataset, CMPD outperforms all baseline with F_1 of 0.971.

References

1. Aliero, M.S., Ghani, I., Qureshi, K.N., Rohani, M.F.: An algorithm for detecting SQL injection vulnerability using black-box testing. *J. Ambient Intell. Human. Comput.* **11**(1), 249–266 (2019). <https://doi.org/10.1007/s12652-019-01235-z>
2. Atlidakis, V., Godefroid, P., Polishchuk, M.: Rest-ler: automatic intelligent REST API fuzzing. *CoRR* abs/1806.09739 (2018)

3. Deepa, G., Thilagam, P.S., Khan, F.A., Praseed, A., Pais, A.R., Palsetia, N.: Black-box detection of XQuery injection and parameter tampering vulnerabilities in web applications. *Int. J. Inform. Secur.* **17**(1), 105–120 (2017). <https://doi.org/10.1007/s10207-016-0359-4>
4. Deepa, G., Thilagam, P.S., Praseed, A., Pais, A.R.: Detlogic: a black-box approach for detecting logic vulnerabilities in web applications. *J. Network Comput. Appl.* **109**, 89–109 (2018). <https://doi.org/10.1016/j.jnca.2018.01.008>
5. Ed-Douibi, H., Izquierdo, J.L.C., Cabot, J.: Automatic generation of test cases for REST APIs: a specification-based approach. In: 22nd IEEE International Enterprise Distributed Object Computing Conference, EDOC 2018, 16–19 Oct 2018, pp. 181–190. Stockholm, Sweden. IEEE Computer Society (2018). <https://doi.org/10.1109/EDOC.2018.00031>
6. Ingham, K.L., Inoue, H.: Comparing anomaly detection techniques for HTTP. In: Kruegel, C., Lippmann, R., Clark, A. (eds.) RAID 2007. LNCS, vol. 4637, pp. 42–62. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74320-0_3
7. Liang, J., Zhao, W., Ye, W.: Anomaly-based web attack detection: a deep learning approach. In: Proceedings of the VI International Conference on Network, Communication and Computing, ICNCC 2017, 8–10 Dec 2017, pp. 80–85. ACM, Kunming, China (2017). <https://doi.org/10.1145/3171592.3171594>
8. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: learning to detect malicious web sites from suspicious urls. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28–July 1, 2009, pp. 1245–1254. ACM, Paris, France (2009). <https://doi.org/10.1145/1557019.1557153>
9. Mac, H., Truong, D., Nguyen, L., Nguyen, H., Tran, H.A., Tran, D.: Detecting attacks on web applications using autoencoder. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, SoICT 2018, 06–07 Dec 2018. pp. 416–421. ACM, Danang City, Vietnam (2018). <https://doi.org/10.1145/3287921.3287946>
10. Nguyen, H.T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., Franke, K.: Application of the generic feature selection measure in detection of web attacks. In: Herero, Á., Corchado, E. (eds.) CISIS 2011. LNCS, vol. 6694, pp. 25–32. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21323-6_4
11. Rieck, K., Laskov, P.: Detecting unknown network attacks using language models. In: Büschkes, R., Laskov, P. (eds.) DIMVA 2006. LNCS, vol. 4064, pp. 74–90. Springer, Heidelberg (2006). https://doi.org/10.1007/11790754_5
12. Sudhodanan, A., Armando, A., Carbone, R., Compagna, L.: Attack patterns for black-box security testing of multi-party web applications. In: 23rd Annual Network and Distributed System Security Symposium, NDSS 2016, 21–24 Feb 2016. The Internet Society, San Diego, California, USA (2016). <http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/attack-patterns-black-box-security-testing-multi-party-web-applications.pdf>
13. Vartouni, A.M., Kashi, S.S., Teshnehlab, M.: An anomaly detection method to detect web attacks using stacked auto-encoder. In: 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), pp. 131–134 (2018)
14. Vigliani, E., Dallago, M., Ceccato, M.: RESTTESTGEN: automated black-box testing of restful apis. In: 13th IEEE International Conference on Software Testing, Validation and Verification, ICST 2020, 24–28 Oct 2020, pp. 142–152. IEEE, Porto, Portugal (2020). <https://doi.org/10.1109/ICST46399.2020.00024>

15. Wang, J., Zhou, Z., Chen, J.: Evaluating CNN and LSTM for web attack detection. In: Proceedings of the 10th International Conference on Machine Learning and Computing, ICMLC 2018, 26–28 Feb 2018, pp. 283–287. ACM, Macau, China (2018). <https://dl.acm.org/citation.cfm?id=3195107>
16. Yu, H.: A Study of Key Techniques for Fuzz Testing of Restful API Interfaces, pp. 1–72. Southeastern University (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Webpage Tampering Detection Method Based on BiGRU-CRF-RCNN

Xiangyu Fan, Jilong Yang, Wei Zhao, Jincheng Deng^(✉), and Fangming Liu

Beijing Knownsec Information Technology Co.,Ltd, Beijing 100097, China
{fanxy, yangjl, zhaow, dengjc, liufm}@knownsec.com

Abstract. With the development of the Internet, cyber security events occur frequently, especially webpage tampering events account for a high proportion. In response to this phenomenon, this paper constructs a webpage tampering detection framework BCR. Based on the webpage to be detected, the webpage text data is segmented and extracted according to the webpage structure, the text features are extracted by using BiGRU model combined with context dependence, and then combined with the CRF to learn sequence state labeling named entities, the word vector is constructed by the extracted named entity and brought into the RCNN model for tampering detection. The experiment results show that the framework has achieved 95.37% precision, 95.35% recall and 95.34% F1-Score in webpage tampering detection, which is better than Textrank RCNN framework in webpage tampering detection. In practical application, it also achieved 95.13% precision and 93.25% recall.

Keywords: Webpage tampering · Named entity recognition · Text classification · Bidirectional gated cyclic unit network · Conditional random field

1 Introduction

With the rapid development of the Internet, various cyber security incidents continue to occur, among which the proportion of webpage tampering events has always been high. How to quickly and accurately locate the tampered content in the webpage and rectify it in time is of great significance to reducing the loss of the site.

At this stage, NLP technology is developing rapidly, text classification technology has a wide range of applications in various fields, and named entity recognition technology is becoming more and more mature. This paper is based on the named entity model to extract the named entities of the text in the webpage segment by segment, and then combined with the text classification model to identify the tampered text.

2 Research Status

At present, the commonly used webpage tampering detection methods are mainly through image recognition and comparison and rule-based detection. Yan Yufeng and

Shen Yong [1] proposed to capture the original image and real-time image of the webpage and detect the feature point information in the before and after images according to the image processing model, and calculate the similarity of webpages according to the feature point information to determine whether the webpage has been tampered with. This method has a good application in the detection of webpage tampering with relatively fixed content or low content update frequency, however, in the case of webpage tampering detection with high content update frequency and rich content, it will affect the model efficiency and detection accuracy. Hongwei R et al. [2] proposed to classify webpage attributes according to principal component analysis, and introduce corresponding rules for each category to realize the judgment of webpage tampering. This method has better effect and efficiency in the scenario of simple webpage structure, but the recognition accuracy will be affected when the web page attributes are complex and the rules cannot cover new objects.

Named entity recognition is a popular research direction of NLP, and named entity recognition models have very good applications in big data research in many fields. The early named entity recognition mainly used the method of building a dictionary, which required a lot of labor costs. After continuous optimization and iteration, today's named entity recognition model mainly relies on various machine learning algorithms to achieve. In the field of named entity recognition in cyber security, Chiu J et al. [3] proposed a method of combining BiLSTM-CNN to build a dictionary in a neural network to encode some words and then match them, this method has better F1-Score than other methods on open source datasets. Fan Xiaoxia et al. [4] proposed a method of constructing a named entity recognition system (DNER) for darknet market text based on Branwen's open source darknet market data text using CBOW-CNN-BiLSTM-CRF. Of entity types, the system can significantly improve recognition. Yi F et al. [6] proposed a named entity recognition model based on regular expressions, entity dictionary, CRF combined with feature templates after considering the particularity and complexity of security entities, got good results.

3 Research Content and Methods

It can be seen from the above that most of the detection of webpage tampering, the final data carrier is text data, how to extract effective and well-characterized key words from the text data plays a decisive role in webpage tampering detection. Different webpages have different text complexity, there is often more noise text data in complex text, and the structure of complex text is more complex than simple text, which has a great impact on the extraction of key words with effective features. In view of the interference of complex text data, this paper designs and implements a framework that extracts text data segment by segment according to the structure of webpages, and then uses named entity model to extract named entities to construct text vectors and bring them into the text classification model for webpage tampering detection, including: Data Preprocessing Framework, BiGRU-CRF Named Entity Recognition Model, RCNN text classification model.

3.1 Data Sources

The experimental data in this paper comes from the historical data of webpage tampering monitoring in the threat intelligence data of Knownsec Security Intelligence Brain. The data is HTML text data, involving five types of websites of government, universities, hospitals, transportation, and energy. It contains 20,000 untampered webpage data and 10,000 tampered webpage data. The tampered content involves pornography, gambling, novels, tripartite movie website, tripartite investment website and reactionary information.

3.2 Data Preprocessing

According to the above content and method, the original data is firstly extracted in segments according to the structure of the webpage, and then perform manual labeling and stop word filtering on the extracted data.

Data Extraction. 1) Parse the HTML data. 2) Build a DOM tree. 3) Traverse the DOM tree to find the tag where the required text is located. 4) Extract the text data segmented based on the webpage structure from the returned HTML data according to the tag.

Data Labeling

1) Named Entity Labeling

This paper uses the word segmentation tool Jieba to perform word segmentation and part-of-speech tagging on the text data. Since named entities are derived from nouns, data labeling is based on the nouns after word segmentation. According to the tampering content of the webpage, a total of 5 types of entity types are labeled, including: PER (person), ORG (company/organization), PLF (platform), OBJ (special noun), 0 (irrelevant word), to ensure that each segment corresponds to one Named Entity Labeling to serve as the data basis for subsequent model building.

2) Text Classification Labeling

According to whether it has been tampered or not, the text category is labeled as 0 (not tampered) and 1 (tampered).

3) Label the page to which the text belongs

Use each webpage domain name as the source label of segmented text data to facilitate subsequent positioning.

Stop Word Filtering. Build a stop word database, including: webpage navigation vocabulary, website copyright statement vocabulary, common auxiliary words, special symbols, etc.

3.3 Text Vectorization

Use word2vec to build text vectors. Word2vec has two models of CBOW and SKIP-GRAM in building text vectors. The CBOW model predicts the central word according to the context of the input text, and the SKIP-GRAM model predicts the context according to the central word. Based on the research background, this paper adopts the CBOW model to construct text vectors.

3.4 BiGRU Model

In the field of named entity recognition, the LSTM model has a wide range of applications. In the LSTM model, a single module consists of three gate units: input gate, forget gate, and output gate. The input gate determines the necessary information to retain, the forget gate determines to discard the information, and the output gate shows the final result. In the GRU network, the three gating units of the LSTM model are replaced by the update gate and the reset gate. The update gate determines the amount of attention information, and the reset gate determines the amount of forgotten information. The reduction of gating units also reduces the parameters in the network, making GRU more concise and efficient than LSTM. BiGRU is a neural network model composed of two unidirectional and opposite GRUs, The current hidden layer state of BiGRU is jointly determined by the current input X_t , the forward hidden layer state h_{t-1}^{\rightarrow} at time $t - 1$, and the backward hidden layer state h_{t-1}^{\leftarrow} at time $t - 1$. The state of the hidden layer at time t :

$$h_t^{\rightarrow} = G(X_t, h_{t-1}^{\rightarrow}) \quad (1)$$

$$h_t^{\leftarrow} = G(X_t, h_{t-1}^{\leftarrow}) \quad (2)$$

$$h_t = \omega_t h_t^{\rightarrow} + \vartheta_t h_t^{\leftarrow} + b_t \quad (3)$$

The function $G()$ is a nonlinear transformation of the input word vector, encoding the word vector at this moment into the corresponding hidden layer state, ω_t and ϑ_t respectively represent the weights corresponding to h_t^{\rightarrow} and h_t^{\leftarrow} at time t , and b_t represents the corresponding bias. Its structure diagram is shown in Fig. 1:

3.5 CRF Model

The Conditional Random Field (CRF) model is a special Markov random field. It is assumed that there are only observation values X and state values Y in the model. In the CRF model, each state value Y_n is only related to its adjacent state value, and its observation value X_n is not has Markov properties. The CRF model needs to consider the correlation between the output state values. The feature function ∂ can be used to learn the relationship between states. The CRF will output a sequence score, and normalize all sequence scores to find the path with the highest probability as the prediction sequence. The CRF model includes state feature function ∂ and state transition function μ .

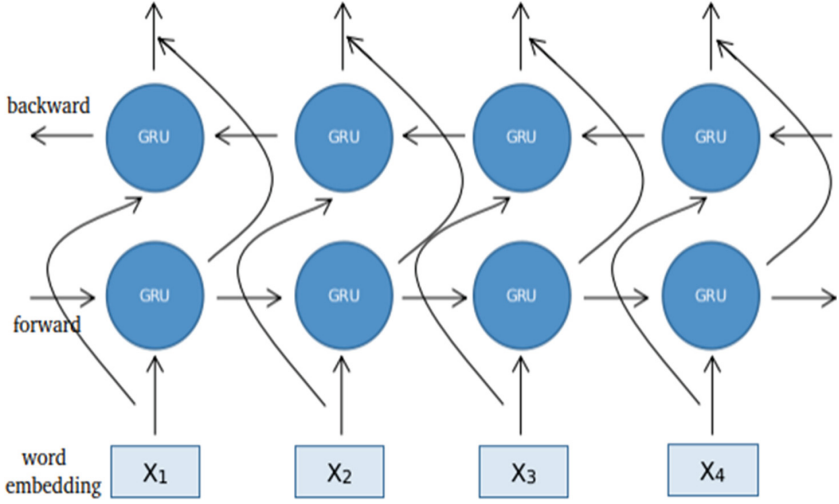


Fig. 1. BiGRU model structure diagram

State Feature Function. Only related to the current node, ϑ represents the current weight of the feature function, that is: $\vartheta \partial(Y_i, X_i)$.

State Transition Function. Related to both node $i + 1$ and node $i - 1$, ω represents the current weight of the transfer function, that is: $\omega \mu(Y_{i+1}, Y_{i-1}, Y_i, X_i)$.

Suppose there are state feature functions $\partial_1, \partial_2, \dots, \partial_L$ whose weights are $\vartheta_1, \vartheta_2, \dots, \vartheta_L$, and transition state feature functions $\mu_1, \mu_2, \dots, \mu_K$, whose weights are $\omega_1, \omega_2, \dots, \omega_L$, for the sequence $X = \{X_1, X_2, \dots, X_n\}$, the probability of the output sequence Y can be calculated as:

$$P(Y|X) = \frac{1}{Z(X)} \exp\left(\sum \vartheta_L \partial_L(Y_i, X_i) + \sum \omega_K \mu_K(Y_{i+1}, Y_{i-1}, Y_i, X_i)\right) \quad (4)$$

of which:

$$Z(X) = \sum \exp\left(\sum \vartheta_L \partial_L(Y_i, X_i) + \sum \omega_K \mu_K(Y_{i+1}, Y_{i-1}, Y_i, X_i)\right) \quad (5)$$

$Z(X)$ is the generalization factor, which can be seen as the sum of the scores of all output sequences.

When the transition feature and state feature are represented by unified functions s and f , the probability of the output sequence Y is:

$$P(Y|X) = \frac{1}{Z(X)} \exp \sum s_i f_i(Y, X) \quad (6)$$

of which:

$$Z(X) = \sum \exp \sum s_i f_i(Y, X) \quad (7)$$

When the CRF model is used for named entity recognition, its graph structure is shown in Fig. 2:

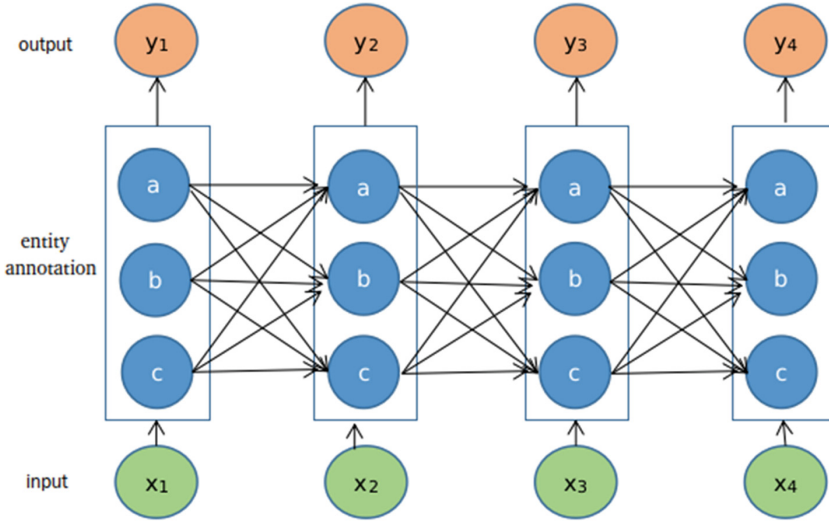


Fig. 2. CRF model structure diagram

3.6 RCNN Model

The RCNN model is a commonly used text classification model, and its structure is divided into three parts.

Region-CNN Model. A bidirectional RNN model is used to obtain the context information of each word embedding, and its expression is:

$$c_l(w_i) = f(W_{(l)}c_l(w_{i-1}) + W_{(sl)}e(w_{i-1})) \tag{8}$$

$$c_r(w_i) = f(W_{(r)}c_r(w_{i+1}) + W_{(sr)}e(w_{i+1})) \tag{9}$$

of which:

$c_l(w_i)$ represents the above of the word w_i .

$c_r(w_i)$ represents the context of the word w_i .

$e(w_i)$ represents the embedding vector of word w_i .

$W_{(l)}$ and $W_{(r)}$ are weight matrices, which transfer the above and below of the previous word to the above and below of the next word.

$W_{(sl)}$ and $W_{(sr)}$ are feature matrices, which combine the semantic features of the current word to the upper and lower parts of the next word.

Computing Hidden Semantic Vectors. The context information obtained in the previous step is merged with the expanded word embedding information, and the activation function is used to calculate the hidden semantic feature vector of the word w_i . Expanded word embedding information is:

$$X_i = [c_l(w_i); e(w_i); c_r(w_i)] \tag{10}$$

Hidden semantic vector is:

$$Y_i^{(2)} = \tanh\left(W^{(2)}X_i + b_{(2)}\right) \tag{11}$$

Continuous Learning, Output Results. After continuous learning of TextCNN, max-pooling and fully connected layers, the classification result is obtained.

The structure diagram of the RCNN model is shown in Fig. 3:

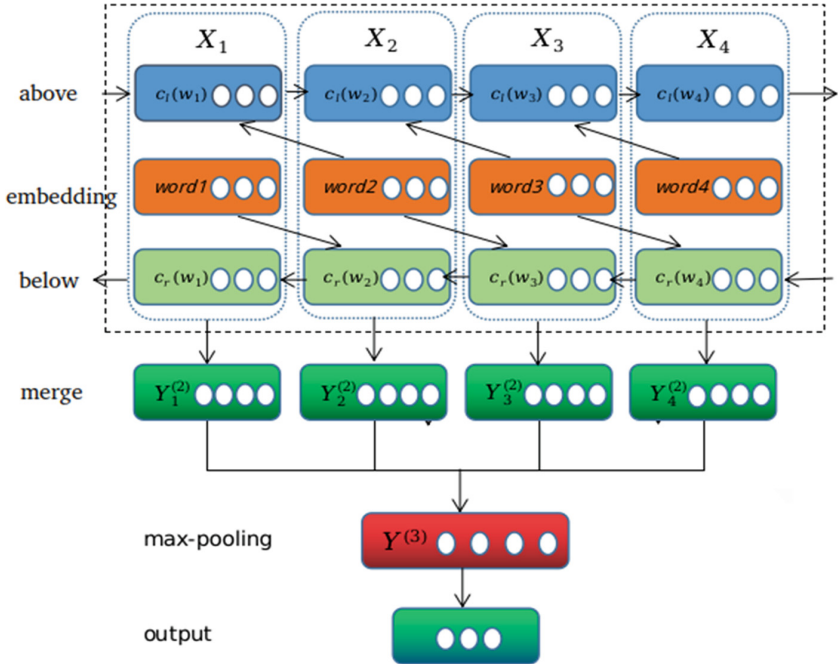


Fig. 3. RCNN model structure diagram

4 Experiment and Result Analysis

4.1 Experimental Environment and Evaluation Indicators

This experiment was performed in the following configuration:

In this experiment, both the named entity model and the text classification model use the precision rate (PRE), the recall rate (REC), and the comprehensive evaluation (F1-Score) as the model’s accuracy evaluation indicators.

4.2 Experimental Configuration

Named Entity Recognition. The 30,000 pieces of data after data preprocessing are divided into training set, test set and validation set according to the ratio of 6:2:2. The distribution of the data set is as follows:

In order to verify that the framework proposed in this paper is better, BiGRU-CRF model, BiLSTM-CRF model, and CNN-LSTM model are set up as comparison models. The three comparison model structures are shown in Table 3 (Tables 1 and 2):

Table 1. Configuration table.

Software and hardware	Configuration
CPU	i7-6700HQ @2.6 GHz
GPU	GTX 970 m
Memory	16 GB
Operating System	Deepin 20.5 GNU/Linux

Table 2. Named entity dataset partitioning.

Data set	Quantity (bar)
Training set	18000
Test set	6000
Validation set	6000

Table 3. Named entity vs model structure.

	BiGRU-CRF	BiLSTM-CRF	CNN-LSTM
Layer1	Input	Input	Input
Layer2	Embedding	Embedding	Embedding
Layer3	bgru	blstm	conv
Layer4	dense	dense	lstm
Layer5	crf_dense	crf_dense	dropout
Layer6	crf	crf	time_distributed
Layer7	–	–	activation

The main parameter configuration of each model is shown in Table 4 (Table 5):

Text Categorization. The 30,000 pieces of data after data preprocessing are divided into training set, test set and validation set according to 6:2:2. The distribution of the data set is as follows:

Table 4. Parameter configuration.

Model	Layer	Epoch	Batch_size	Active
BiGRU-CRF	bgru	30	256	tanh
BiLSTM-CRF	blstm	30	256	tanh
CNN-LSTM	lstm	30	256	softmax

Table 5. Text classification dataset partitioning.

Data set	Quantity (bar)
Training set	18000
Test set	6000
Validation set	6000

Use two methods to build word vectors and then bring them into the RCNN model for comparison. They are: Named entities combined with RCNN model for classification, Text summarization combined with RCNN model for classification. The RCNN model epoch is set to 30, batch_size is set to 256, and the training process is shown in Table 6:

Table 6. RCNN model training process

Layer	Output shape	Active
input	(None, 12)	–
layer_embedding	(None, 12, 100)	–
layer_conv1d	(None, 8, 128)	relu
layer_max_pooling	(None, 128)	–
layer_dense	(None, 64)	relu
dense_1	(None, 2)	softmax

4.3 Experimental Results and Analysis

Named Entity Recognition. The accuracy indicators of each model are shown in Fig. 4:

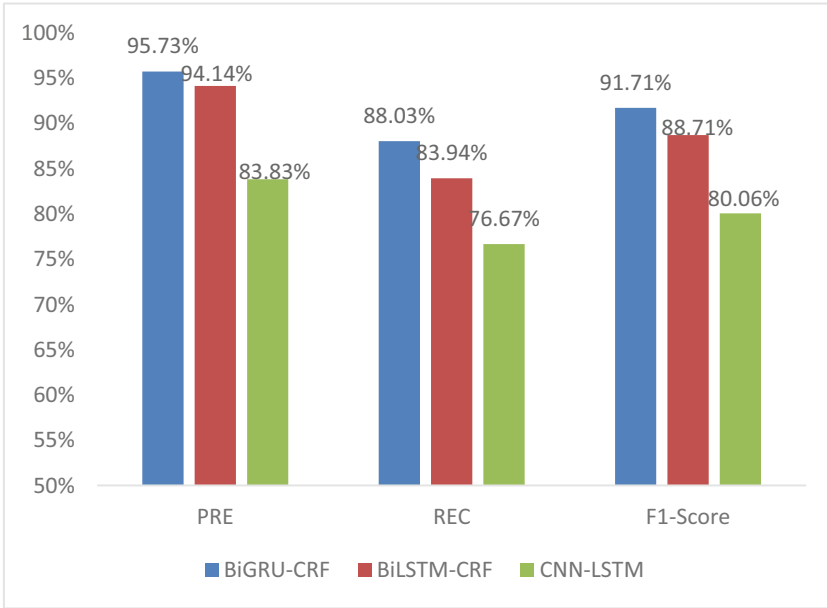


Fig. 4. Accuracy indicators of each named entity model

In terms of recognition accuracy, the PRE, REC, and F1-Score of the BiGRU-CRF model in this scenario are 93.88%, 91.36%, and 92.60% respectively, which is a certain improvement compared to the other two models. The main reason is that the data set is based on segmented text data after webpage structure segmentation, and the BiGRU-CRF model has improved and optimized the gate control unit compared with the BiLSTM-CRF model, and has better applications in simple text data. Both BiGRU-CRF model and BiLSTM-CRF model can encode text information from front to back and from back to front, which can better capture bidirectional text semantic dependencies, while CNN-LSTM model cannot encode text information from back to front, It can only capture one-way text semantic dependencies, so it is lower than the other two models in terms of accuracy.

Figure 5, Fig. 6, and Fig. 7 show the evaluation indicators of each category of named entity recognition accuracy of each model:

Compared with the other two models, the BiGRU-CRF model has obvious advantages in PLF named entity recognition, and is comparable to the BiLSTM-CRF model in other types of named entity recognition. The CNN-LSTM model is far behind the other two models in terms of OBJ and PLF named entity recognition. From the comprehensive view of the above radar charts, BiGRU-CRF is relatively better in named entity recognition in this scenario.

Text Categorization. The accuracy evaluation indicators of each model are shown in Fig. 8:

Compared with TextRank-RCNN, BiGRU-CRF-RCNN has a certain improvement in precision, recall and F1-Score. The main reason is that BCR framework extracts

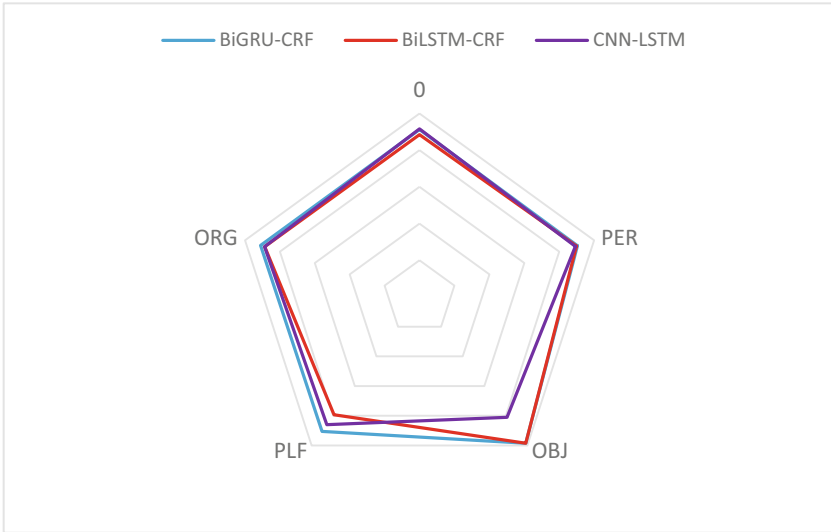


Fig. 5. The precision of each model for each type of named entity recognition

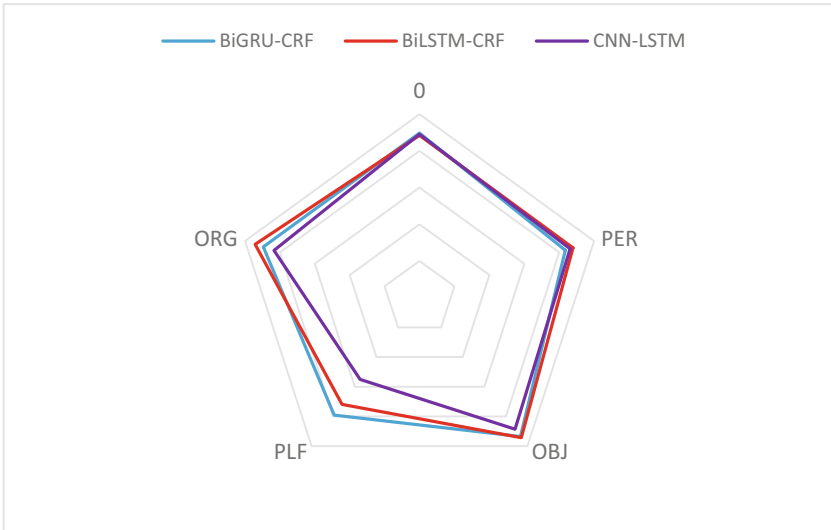


Fig. 6. The recall of each model for each type of named entity recognition

keywords representing text based on the characteristics of BiGRU-CRF model. Entities can better represent the domain features and context features of the current text. While the TextRank-RCNN framework constructs a network based on the relationship between local adjacent nodes when extracting keywords representing text. The mechanism of exclusive nouns, the extracted information features are not comprehensive, so the accuracy of tampering identification is relatively poor.

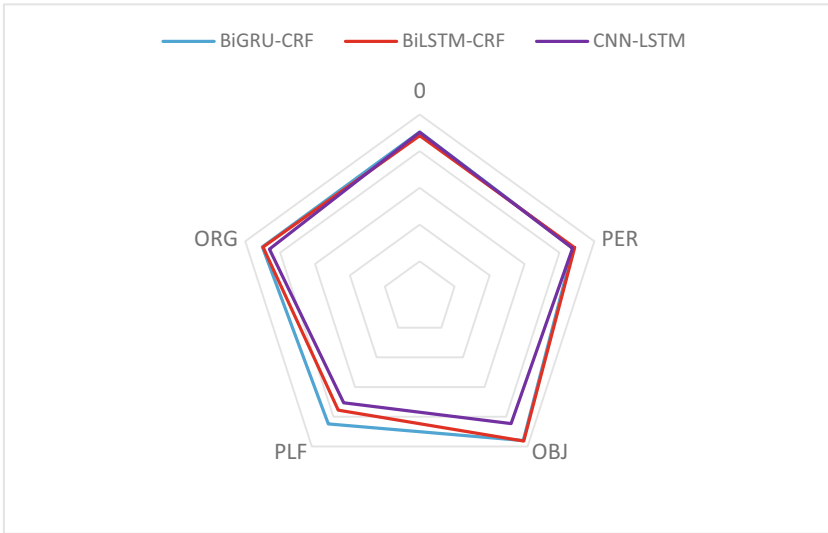


Fig. 7. Each model recognizes the F1-Score for each type of named entity

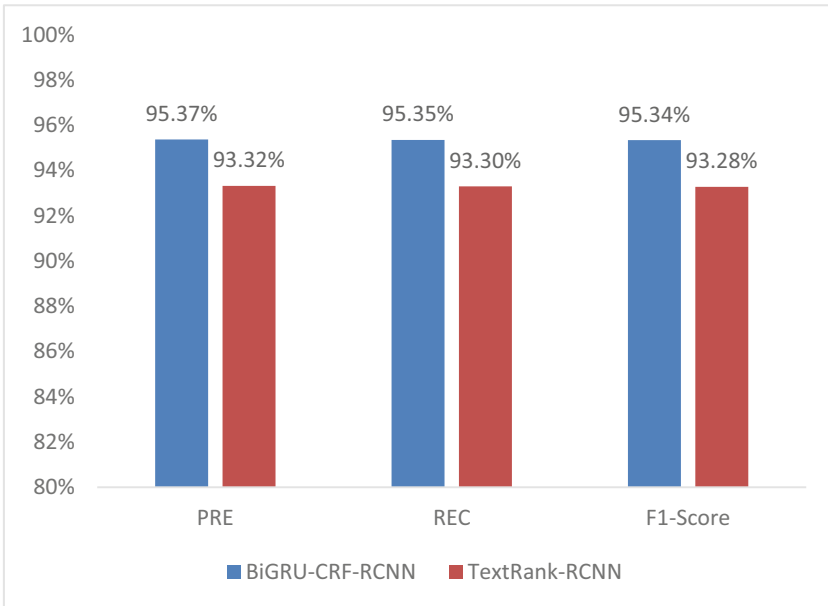


Fig. 8. The accuracy index of each text classification model

4.4 Practical Application

This framework has been applied in Knownsec Security Intelligence Brain. From the test results, an average of 108,326 webpages are detected every day, and an average of

411 tampered webpages are identified every day. After manual sampling by the sampling team, the sampling precision was 95.13%, and the recall was 93.25%.

4.5 Conclusion

At this stage, named entities and text classification technology have been widely used in the field of cyber security, but less in webpage tampering detection. Therefore, the BiGRU-CRF-RCNN framework is proposed for webpage tampering detection. According to the above experimental process and practical application effect, we can get:

Advantages of this Framework. Due to the structural characteristics of the gated unit of the BiGRU-CRF model, it has a better application than other models in this scenario. In terms of text classification, the named entities extracted based on the named entity model can better reflect the characteristics of the current field. Therefore, in the scenario of this paper, using the text vector constructed based on named entities for text classification has a better effect.

Weaknesses of the Framework. The BiGRU-CRF-RCNN model achieves better results because the industry content of the website detected in production and experiments is less related to the tampered content. Considering the problem of model generalization, if the data surface is widened, and the positive samples and negative samples are related, it needs to be improved according to the actual effect.

References

1. Yan, Y., Shen, Y.: Web page tamper detection based on image processing. *Comput. Digit. Eng.* **48**(6), 5 (2020)
2. Hongwei, R., Liu, T., Hua, L., et al.: Webpage tamper detection based on principal component analysis. *China Sci. Pap.* (2012)
3. Chiu, J., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. *Comput. Sci.* **4**, 357-370 (2015)
4. Fan, X., Zhou, A., Zheng, R., et al.: Darknet market named entity recognition based on deep learning. *J. Inf. Secur. Res.* (2021)
5. Qin, Y., Shen, G., Zhao, W., et al.: Research on the method of network security entity recognition based on deep neural network. *J. Nanjing Univ. Natl. Sci.* **55**(1), 12 (2019)
6. Yi, F., Jiang, B., Wang, L., et al.: Cybersecurity named entity recognition using multi-modal ensemble learning. *IEEE Access*, (99), 1-1 (2020)
7. Li, K.-Y., Liu, X.-D.: Web tampering detection system based on feature recognition. *Electron. Des. Eng.* **28**(440)(18), 22-25+30 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Cryptocurrency



Detecting Bitcoin Nodes by the Cyberspace Search Engines

Ruiguang Li^{1,2(✉)}, Jiawei Zhu², Jiaqi Gao³, Fudong Wu³, Dawei Xu^{1,3},
and Liehuang Zhu¹

¹ School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
lrg@cert.org.cn

² National Computer Network Emergency Response Technical Team/Coordination Center of
China, Beijing, China

³ School of Cyber Security, Changchun University, Changchun, China

Abstract. Nowadays, the cyberspace search engines have showed great power to find entities and services in the network, which provide new ideas and methods to detect the Bitcoin nodes. This paper introduces the Bitcoin's P2P network and nodes including the reachable nodes and the unreachable nodes. Then, the results of detecting reachable nodes by the cyberspace search engines are showed. Next, the author proposes a new approach to find and verify the unreachable nodes by the cyberspace search engines. Finally, this paper illustrates the de-anonymization of some Bitcoin nodes by the cyberspace search engines, which map some node's IP addresses to real Bitcoin entities, such as Zeblockchain (a browser website), Microwallet (a wallet website) and Laurentia Pool (a non-profit pool website).

Keywords: Cyberspace search engines · Bitcoin nodes · Reachable nodes · Unreachable nodes · De-anonymization

1 Introduction

The cyberspace search engine is a new kind of network tool, which has attracted more and more attention from network researchers in recent years. It is different from the traditional Web search engine which takes Web pages as the retrieval objects, such as Google, Baidu and Bing. The Web search engine is widely crawling and storing web pages in the network, extracting and analyzing the pages' content, and providing keyword retrieval services for the public. The cyberspace search engine finds the entities and services in the network by actively detecting, obtains the target's information through protocol interaction and makes a comprehensive display. At present, the well-known cyberspace search engines in the industry include: Shodan (shodan.io, US), Censys (censys.io, US), BinaryEdge (www.binaryedge.io, EU), Zoomeye (www.zoomeye.org, CN), Fofa (offline now, CN), and so on.

The cyberspace search engines commonly maintain a protocol library which contains a variety of protocols, deploy probes all around the world, detect the whole network using various protocols, and find open ports and services all the time. There are many kinds of

equipments in the network, including Servers, Network equipments, Terminals, Office facilities, Smart home devices, Industrial controlling equipments, Webcams, Blockchain entities, etc. [1] made a detailed comparative analysis of the well-known cyberspace search engines, compared their supporting protocols, detected equipments, equipment types, detecting capabilities, system structure, and probes, etc.

Bitcoin is the most successful electronic crypto-currency in the world. It was proposed by Nakamoto in 2008 [2] and launched officially in January 2009. Bitcoin kept running stably since then and had become an important means for global finance and payment. The cyberspace search engines had strong infrastructures which offer powerful computing, abundant storage, and a large amount of detecting records. This gave a great convenience for the analysis of assets and equipments in the cyberspace. The well-known cyberspace search engines include Shodan, Censys, Zoomeye, Fofa, etc, which provide detecting services for Bitcoin nodes. In this paper, we will introduce our work of finding and analyzing Bitcoin nodes by cyberspace search engines.

The contributions of this paper are as follows: 1) Introduce the results of detecting the Bitcoin reachable nodes by the cyberspace search engines. 2) Propose a new approach to find and verify the Bitcoin unreachable nodes by the cyberspace search engines. 3) Illustrate the de-anonymization of some Bitcoin nodes by the cyberspace search engines, which map some node's IP addresses to real Bitcoin entities.

2 Bitcoin Network and Nodes

Bitcoin system can be logically divided into the network layer and the transaction layer, as shown in Fig. 1.

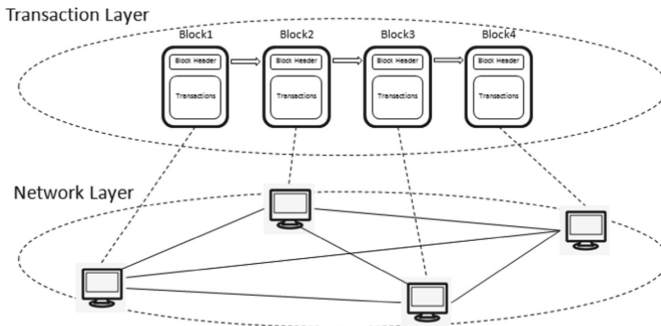


Fig. 1. Bitcoin's logic structure

The network layer is composed of a large number of Bitcoin nodes. Each node keeps working on broadcasting IP addresses, verifying transactions, packaging blocks, and mining independently. All the transactions are stored in the blocks, which connected each other by time order to form a blockchain. All the transactions are published to all network participants and stored in all nodes of the network. Previous studies mostly focused on the transaction layer, but less on the network layer. The Bitcoin network is a typical P2P

network without an organization or trust center. Nodes gain trust between each other through interactions, and form the network by themselves. The Bitcoin network is the foundation of Bitcoin system.

The Bitcoin network can be divided into the visible part (reachable nodes) and the invisible part (unreachable nodes). The reachable nodes can receive incoming connections and provide public services for the whole network. Generally, they would store a complete copy of the blockchain data. They open fixed ports (often 8333) waiting for connections and can be regarded as “Servers” in the network.

The unreachable nodes do not receive incoming connections from outside and don't provide public services for the whole network. They don't keep a complete copy of the blockchain data and can be regarded as “clients”. The unreachable nodes are generally deployed behind a NAT (Network Address Translation) or a firewall and cannot be found by active detecting. Bitcoin nodes have different connectivity, service type, and topology, which have great impacts on the performance of Bitcoin system. Therefore, it is important to detect and study the Bitcoin nodes in depth.

3 Detecting the Reachable Nodes

Because the reachable nodes open fixed ports waiting for outside connections, we can easily detect all the reachable nodes by active detecting. There were many studies by far. Joan et al. [3]. Measured the Bitcoin network from November 2013 to January 2014, connected the Bitcoin nodes with bitcoin-sniffer (a open source tool) [4], collected 872000 nodes, and analyzed the nodes' geographical distribution, stability, propagation delay, etc. Christian Decker et al. [5] in 2013 and Giuseppe Pappalardo et al. [6] in 2016 measured the Bitcoin network, observed the propagation delay of blocks and transactions in the network. In the same year, Fadhil et al. measured the Bitcoin network for a week and collected 6430 stable online nodes and 313676 client IP addresses [7]. Sehyun Park et al. carried out a comparative study [8] in 2018, developed a software Bitcoin-Node-Scanner, obtained and verified 1 million nodes' IP addresses within 37 days, and counted the IP types (IPv6/IPv4/Onion), geographical distribution, port numbers, client versions, protocol versions, etc.

All these studies above were carried out by individual researchers. The cyberspace search engines have far more power than the single terminals. By scanning the whole network with probes all over the world, they can connect to all reachable nodes, get their information and make a time-based cumulative analysis. Fofa showed 56748 Bitcoin reachable nodes detected from December 2016 to September 2021 [9]. Zoomeye showed 63504 Bitcoin reachable nodes [10] and display their information such as IP, open ports, open services, countries, affiliated enterprises, protocol slogans, geographic longitude and latitude, as shown in Fig. 2 below.

It should be noted that [9] and [10] are only reachable nodes detected by the cyberspace search engines. Next, we will propose an approach to find and verify unreachable nodes by the cyberspace search engines.

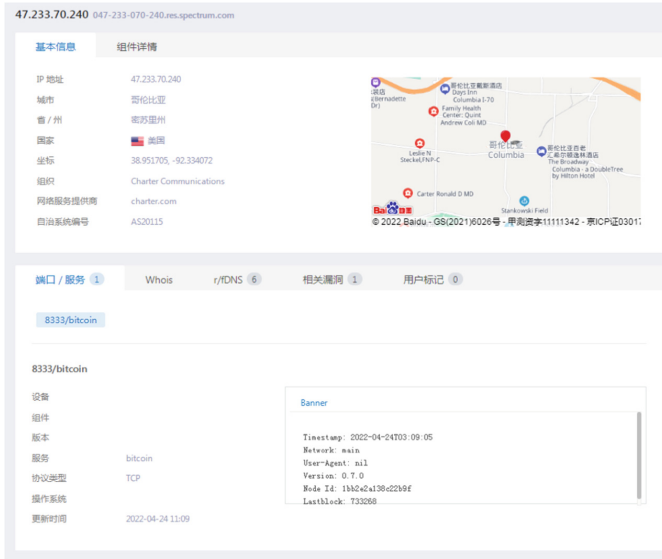


Fig. 2. Zoomeye's page of a reachable node

4 Inferring the Unreachable Nodes

The unreachable nodes don't open ports for outside connections, so they cannot be found by active detecting. Even if we got some addresses of the unreachable nodes, we could not definitely verify them by active detecting due to the existence of "Churn Nodes" which caused by the network delay.

There were a few studies on the unreachable nodes. Alex et al. proposed a de-anonymization method for the unreachable nodes [12] in 2014, which setup some probes connected to all entry nodes. When an unreachable node broadcasted a connection request through the entry nodes, the request would be forwarded to the probes and be recorded. The author believed that there were about 90000 unreachable nodes at that time. Till et al. simulated the Bitcoin network in 2016 and analyzed the broadcasting of transactions in the network [13]. It was estimated that the total number of was about 16000 then. Liang et al. deployed 102 probes around the world to collect the connection requests [14] in 2017, and estimated that there were 155000 unreachable nodes in the whole network. Matthias et al. monitored the "unsolicited" ADDR messages [15] in 2021 and could identify about 31000 active unreachable nodes every day. Federico et al. studied in detail the roles and number of unreachable nodes in Bitcoin network [16], and proposed an improved transactions broadcasting protocol, which improved the efficiency and security of the Bitcoin network. Alex et al. introduced the Bitcoin network based on Tor [17], proposed a man-in-the-middle attack against Bitcoin, and analyzed the delay caused by unreachable nodes in the attack. Indra et al. proposed a de-anonymization method for the unreachable nodes [18] by collecting all GETDATA messages and matching IP addresses with transactions/blocks. The accuracy of identifying the unreachable nodes is up to 90%.

Next, we will propose an approach to infer the unreachable nodes by the cyberspace search engines. First, we setup a fake client to actively connect to the reachable nodes, obtained a large number of Bitcoin nodes' IP addresses by the interaction mechanism of GETADDR-ADDR. Then, we input these addresses to the cyberspace search engine and obtain all the feedback records of the engines. Finally, by analyzing the open services and detecting time of the target IP, we can infer the unreachable nodes. Here we make two judgments.

Judgment 1: If an IP had a record of opening Bitcoin service with a new timestamp (within the duration of detecting cycle), this IP stood for a reachable node.

Judgment 2: If an IP had a record of opening other services (HTTP, SSL, etc.) except for Bitcoin service and the timestamp was relatively new(within the duration of detecting cycle), this IP stood for an unreachable node.

The correctness of Judgment 1 is obvious. The correctness of judgment 2 is also easy to understand. Because if we can verify a real IP from the Bitcoin system is opening other services, but isn't opening the Bitcoin service, the IP must stood for an unreachable node. The worldwide probes and all-weather scanning of the cyberspace search engines made sure that the "unreachable" of nodes were not caused by "network delay". In fact, we have made experiments to testify Judgment 2 and the accuracy was up to 95%.



Fig. 3. Zoomeye's page of a unreachable node

Here is an example. As shown in Fig. 3, the node “167.172.158.149” is a real IP address obtained from the Bitcoin system. We input the IP into Zoomeye and check the feedback records. It can be seen that the IP opened “SSH” and “TCP” service, but didn’t open “Bitcoin” service and the detecting timestamp was “2022-03-24”. So the node “167.172.158.149” is an unreachable node.

To make a Ground-Truth test, we deployed a Bitcoin probe on Vultr. By checking the real neighbors using “peerinfo” command, we found the node “167.172.158.149” was its neighbor and the attribute is “inbound = true”. The node made an incoming connection to our probe and was an real unreachable node.

5 De-anonymization of the Bitcoin Nodes

As an encrypted digital currency, Bitcoin protects the privacy and security of users’ transactions. However, many researchers are very interested in the de-anonymization of Bitcoin addresses and tracing the route of transactions. By far, there are many studies on this issue being published. The existing methods are mainly based on the clustering of transaction addresses. For example, Butian Huang et al. proposed a clustering algorithm “BPC” [19] based on the nodes’ behaviors, which clustered the nodes after behavior similarity measurement. The experiment showed that the accuracy was higher than the previous algorithms. Annika Baumann et al. analyzed the Bitcoin’s transaction graph [20], inferred that there was a close relationship between network usage and exchange rate, and de-anonymized the 11 largest entities in the transaction graph. Meng Shen et al. analyzed the transaction propagation mode, proposed a method to obtain the initial transaction by calculating the pattern matching score [21], and established the association between the transaction and the initiating node’s IP. The experimental accuracy was up to 81.3%.

In the de-anonymization of the Bitcoin nodes, it’s important but difficult to find the association between a node’s IP and the real network entity (exchanges, browsers, wallets or pools), because many important entities keep their IP addresses highly confidential for the reason of privacy. **The cyberspace search engines provide new ideas for the association between Bitcoin nodes’ IP and the real entities.** The cyberspace search engine detect the whole network using various protocols, and will find all services support by a node. For the reachable nodes, all the services such as HTTP, HTTPS, SSL, Bitcoin will be found together. As some persons or companies may open different services in One IP address, we could get extra information for a Bitcoin node by visiting its HTTP page. In some cases, we could get useful information such as the geographic location, organization information, and services operated by the website. Here we gave some examples.

- 1) A node with IP (147.135.252.43). This IP address is a Bitcoin browser “Ze-blockchain”, belonging to Japan Digital Service Company, as shown in Fig. 4.

The screenshot shows the ZeBlockchain website interface. At the top, there are logos and descriptions for Bitcoin, Ethereum, Dash, Litecoin, and Zcash. Below this is a search bar and a section titled "Latest Blocks" with a table of recent transactions.

Height	Age	Transactions	Mined by	Size
754180	8 minutes ago	946		1179207
754179	14 minutes ago	2446	Antifliner	1589780
754178	30 minutes ago	394		497883
754177	32 minutes ago	328		465089
754176	34 minutes ago	716		1547575

Fig. 4. Zeblockchain (a Bitcoin browser)

- 2) A node with IP (216.108.227.39): This IP address is a Bitcoin wallet “Microwallet”, operated by a US company, as shown in Fig. 5.

The screenshot shows the Microwallet website. The main heading is "Accept/Pay Cryptocurrencies free of cost". Below this, there is a sub-heading "MicroWallet is a dedicated service users to store cryptocurrencies and accepting/sending payments instantly." and a "GET STARTED TODAY" button. The background features images of smartphones displaying the Microwallet app interface.

Fig. 5. Microwallet (a Bitcoin wallet)

- 3) A node with IP (51.81.56.49): This IP is a Bitcoin Pool “Laurentia pool”, which is a non-profit mining pool(open source), as shown in Fig. 6.

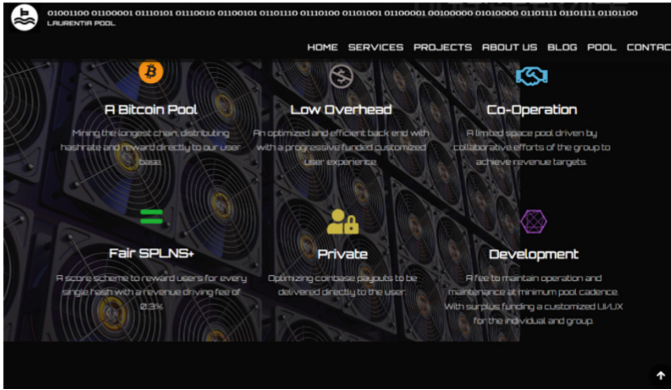


Fig. 6. Laurentia (a Bitcoin pool)

Limitations: This method could only de-anonymize some Bitcoin websites which open different services in One IP address. If large organizations have many IP addresses and don't deploy different services on same IP address, this method is no longer applicable.

6 Summary

This paper introduces the working principle of the cyberspace search engines and discusses their application in detecting the Bitcoin nodes. The Bitcoin network is composed of visible part (reachable nodes) and invisible part (unreachable nodes), which have different characteristics. The reachable nodes provide public services for the network and easy to detect, while the unreachable nodes are only clients and hidden in the network. The number of the unreachable nodes is about ten times to the reachable nodes [14], which are not easy to detect and analyze.

The author introduces the results of detecting Bitcoin nodes by the cyberspace search engines, then proposes a new approach to verify the Bitcoin unreachable nodes, finally illustrates the de-anonymization of the Bitcoin nodes which could find the association between a node's IP and the real network entity (a exchange, a browser, a wallet or a pool). By far, the cyberspace search engines can only detect Bitcoin nodes with Ipv4 addresses, and Ipv6 addresses are not supported. However, with the fast improvement of the cyberspace search engines, they will play more important roles in the detecting and analyzing of the Bitcoin network.

Acknowledgement. This work is supported by National Natural Science Foundation of China (Grant No. 62106060), and National Key Research and Development Program of China (Grant No.2020YFB1006105).

References

1. Li, R., Shen, M., Yu, H., et al.: A Survey on Cyberspace Search Engines. Springer, Singapore. Springer, Singapore (2020)
2. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2009). <http://www.bitcoin.org/bitcoin.pdf>
3. Donet, J.A.D., Pérez-Sola, C., Herrera-Joancomartí, J.: The bitcoin P2P network. In: Proceedings of Financial Cryptography. Data Security FC Workshops, BITCOIN WAHC, vol. 16, pp. 87–102 (2014)
4. Sebicas.: Bitcoin p2p Network Sniffer (2013). <http://github.com/sebicas/bitcoin-sniffer>
5. Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: Proceedings of IEEE 13th International Conference Peer-Computer (P2P), pp. 1_10 (2013)
6. Pappalardo, G., Caldarelli, G., Aste, T.: The Bitcoin Peers Network. http://blockchain.cs.ucl.ac.uk/wpcontent/uploads/2016/11/P2PFISY2016_paper_32.pdf. Accessed 30 Jun 2016
7. Fadhil, M., Owenson, G., Adda, M.: A bitcoin model for evaluation of clustering to improve propagation delay in bitcoin network. In: Proceedings of IEEE International Conference Computer Science Engineering (CSE), IEEE International Conference Embedded Ubiquitous Computing (EUC), 15th International Symposium Distribution Computer Application for Business Engineering (DCABES), pp. 468–475 (2016)
8. Park, S., Im, S., Seol, Y., Paek, J.: Nodes in the bitcoin network: comparative measurement study and survey. IEEE Access **7**, 57009–57022 (2019)
9. <https://fofa.so/>
10. <https://www.zoomeye.org/>
11. <https://bitnodes.io/>
12. Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in bitcoin P2P network. In: Proceedings of ACM SIGSAC Conference Computer Communication Security, pp. 15–29 (2014)
13. Neudecker, T., Andelfinger, P., Hartenstein, H.: Timing analysis for inferring the topology of the bitcoin peer-to-peer network. In: 2016 International IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, Meets the and Smart World (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld), pp. 358–367 (2016) out
14. Liang, W., Pustogarov, I.: Towards Better Understanding of Bitcoin Unreachable Peers (2017)
15. Grundmann, M., Amberg, H., Hartenstein, H.: On the estimation of the number of unreachable peers in the bitcoin P2P network by observation of peer announcements (2021)
16. Franzoni, F., Daza, V.: Improving bitcoin transaction propagation by leveraging unreachable nodes. In: 2020 IEEE International Conference on Blockchain (Blockchain), pp. 196–203. IEEE (2020)
17. Biryukov, A., Pustogarov, I.: Bitcoin over Tor isn't a good idea. In: 2015 IEEE Symposium on Security and Privacy, pp. 122–134 (2015). <https://doi.org/10.1109/SP.2015.15>
18. Mastan, I.D., Paul, S.: A new approach to deanonymization of unreachable bitcoin nodes. In: Capkun, S., Chow, S. (eds.) Cryptology and Network Security. CANS 2017. Lecture Notes in Computer Science, vol. 11261. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02641-7_13
19. Huang, B., Liu, Z., Chen, J., et al.: Behavior pattern clustering in blockchain networks. Multimedia Tools Appl. **76**(19), 20099–20110 (2017)

20. Baumann, A., Fabian, B., Lischke, M.: Exploring the bitcoin network. *WEBIST* (1), 2014, 369–374 (2014)
21. Shen, M., Duan, J., Shang, N., Zhu, L.: Transaction deanonymization in large-scale bitcoin systems via propagation pattern analysis. In: Yu, S., Mueller, P., Qian, J. (eds.) *SPDE 2020. CCIS*, vol. 1268, pp. 661–675. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-9129-7_45

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Research on Bitcoin Anti-anonymity Technology Based on Behavior Vectors Mapping and Aligning Model

Shenwen Lin^(✉), Hongliang Mao, Zhen Wu, and Jinglin Yang

National Computer Network Emergency Response Technical Team/Coordination Center of
China, Beijing, China
lsw@cert.org.cn

Abstract. Traditional anti-anonymity technologies for Bitcoin transactions include two types. One is network-layer anti-anonymity technology, which achieves the purpose of locating the initial IP of specific transaction information by speculating on the IP propagation path of transaction; the other is the anti-anonymity technology of the transaction layer. By analyzing the data of the Bitcoin ledger, it realizes the on-chain behavior portrait of a specific wallet address attributable to the user. In this work, we propose a new anti-anonymity technology, by constructing transaction behavior vectors and social behavior vectors based on Bitcoin ledger data and off-chain social data respectively, and build a model for mapping and aligning the two vectors. Experimental test shows that the proposed anti-anonymity technology is more accurate and has better practical effects. Furthermore, the technology suits for the anti-anonymity of other virtual currencies as well.

Keywords: Bitcoin · Virtual currency · Anti-anonymity · Behavior vector

1 Introduction

Bitcoin is a purely peer-to-peer version of electronic cash [1], which allow online payments to be sent directly from one party to another without going through a financial institution. It relies on digital signatures to prove ownership and a public history of transactions to prevent double-spending. Bitcoin does not rely on third-party credit, has strong anonymity. It mainly reflects three aspects: one is the anonymous transaction address. Bitcoin transaction address is created by the user independently, independent of user identity information, and does not require third-party participation to create and use the address; Second, the fragmented transaction behavior. Bitcoin system supports users to generate different addresses for each transaction. User transaction information can be arbitrarily dispersed in different anonymous address behaviors. Third, the source of Bitcoin transaction package is difficult to find in network. Bitcoin communication network uses P2P protocol, and there is no central node. Transaction information broadcasts all over the network. It is difficult to track the origin of transaction information by

monitoring a single server. Because of its strong anonymity, Bitcoins are often used in gambling, illegal fund-raising, fraud, pyramid sale, money laundering and other illegal activities.

Traditional Bitcoin transaction anti-anonymity technology mainly includes two types: one is the network layer anti-anonymity method, which mainly detects and collects the transaction information broadcast by the Bitcoin network layer, analyzes the propagation path of a specific Bitcoin transaction in the P2P network, infers the IP address of the originating service node of the transaction, and then locates the user IP of the transaction. Another method is the anti-anonymity method at the transaction level, which mainly obtain user portrait information for a specific wallet address by analyzing transaction relationships between different transaction addresses, especially with the help of the labels of the addresses of exchanges, mining pools and other institutions. The above two types of anti-anonymity technologies are not effective because they cannot track the source of the user's social identity information to which the transaction address belongs.

Because of the shortcomings of the traditional anti-anonymity technology of Bitcoin transaction, this paper integrates the data on and off the chain, studies and proposes an anti anonymity technology of Bitcoin transaction based on behavior vector mapping and aligning model. Build a social behavior vector based on off chain social data, and establish a mapping and aligning model with the transaction behavior vector based on Bitcoin ledger data, which can realize the anti-anonymity of Bitcoin address and transaction. Because the social behavior vector contains the real social identity information of users, this paper proposes anti anonymity technology, which has better practical effect than the traditional anti anonymity technology.

2 Bitcoin Transaction Overview

Every transaction in the Blockchain has a list of inputs and outputs, where each includes addresses that were used in the transaction and the amount of coins spent in that transaction. Inputs of the current transaction come from the outputs of the previous transaction, and the output of the current transaction will be used as the input in other transactions, which to form a transaction chain (see Fig. 1).

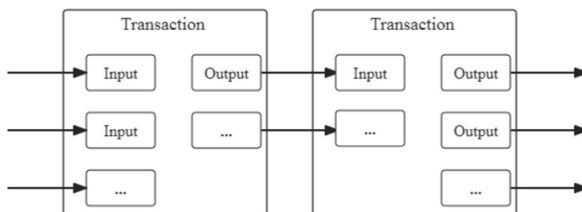


Fig. 1. Bitcoin transaction chain

There will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender, which will be automatically selected by the Bitcoin client as the input in future transactions.

Bitcoin transactions can be roughly divided into two types: the first type is mining reward transactions. Each block has a mining reward transaction. This kind of transaction has no input but only output. The system transfers the mining reward of this block and fee of the transaction contained in the block to the output; The second type is ordinary transactions, including several inputs and several outputs.

Since multiple input addresses of a transaction correspond to different private keys, Bitcoin transferring the input needs the signature of the corresponding private key; Therefore, it is generally believed that multiple input addresses of a transaction belong to the same entity. So, with the help of transaction address clustering, the decentralized transaction behaviors of the same entity in the ledger can be gathered, which is convenient to master the behavior characteristics of the entity.

There are four kinds of transaction address clustering technology [2]. One is the clustering technology based on multiple input addresses. Multiple input addresses of a transaction belong to the same address cluster; The second is the clustering technology based on the change address. The change address of a transaction belongs to the same address cluster as the input address. At the same time, through the change address as the connecting link, the input addresses in the two transactions can be combined into the same address cluster; the third is the clustering technology based on mining reward transaction. Multiple output addresses of a mining reward transaction belong to the same address cluster. The fourth is the comprehensive clustering technology combining the above three clustering technology.

3 Transaction Scene Graph Structure

Bitcoin transaction scene include mining reward, depositor withdrawal on the exchange, gambling, blackmail, MLM fraud, etc. Among them, deposit and withdrawal of Bitcoin on the exchange are more popular.

Deposit transaction transfer Bitcoin held by the user's personal wallet address to the deposit wallet address assigned to the user by exchange. The private key of the deposit wallet address is controlled by the exchange, and different deposit wallet addresses correspond to different users. Deposit transactions include customer to customer (C2C) transaction scene and business to customer (B2C) transaction scene.

The general characteristics of the graph structure of C2C deposit transaction are: a small number of transaction input and two outputs, one of which including user's deposit wallet address, and the cluster label of this address is the name of exchange (see Fig. 2).

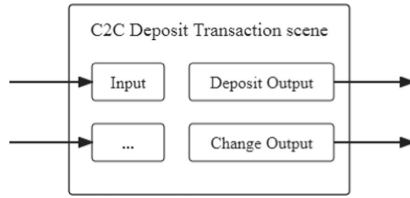


Fig. 2. Graph structure of C2C deposit transaction scene

B2C deposit transaction scene graph has a 1-to-N structure, which is generally characterized by a small number of transaction input addresses and a large number of transaction output, in which the output addresses are deposit wallet addresses of a large number of different users, and the cluster labels of different output addresses are the same or different exchange (see Fig. 3).

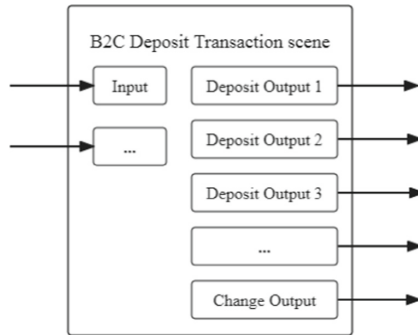


Fig. 3. Graph structure of B2C deposit transaction scene

Withdrawal transaction transfer Bitcoin hosted on the exchange to the wallet address specified by the user. In order to reduce the transaction fee, exchange usually collects multiple users' withdrawal order and transfers Bitcoin to multiple users' wallet addresses in one transaction.

The graph structure of withdrawal transaction has the characteristics of a 1-to-N structure. The cluster labels of transaction input addresses are the same exchange, and the transaction output addresses are specified by a large number of different users (see Fig. 4).

Each transaction needs to pay fee, in reality, there is a combination of deposit transaction and withdrawal transaction, that is, user withdraws Bitcoin on a exchange and deposit it to another exchange.

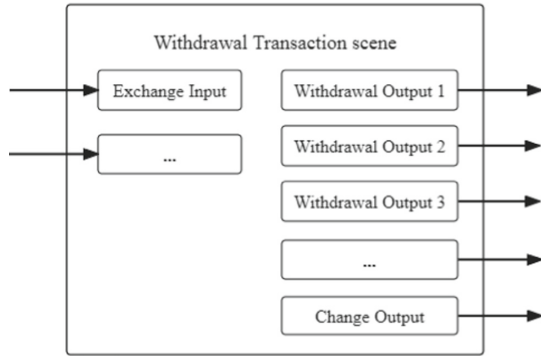


Fig. 4. Graph structure of withdrawal transaction scene

4 Traditional Bitcoin Anti-anonymity Technology

Traditional Bitcoin anti-anonymity technology mainly includes network layer anti-anonymity technology and transaction layer anti-anonymity technology.

Network layer anti-anonymity technology [3] refers to collecting transaction packet transmitted by Bitcoin P2P network, analyzing the propagation path of a specific Bitcoin transaction packet in P2P network, and inferring the server IP of the first broadcast node. For example, koshy et al. [4] used special transactions to find the originating node. Most normal transactions will be forwarded once by multiple nodes, while transactions with wrong format will only be forwarded once by the originating node. Therefore, this feature can be used to identify the originating node of special transactions. However, due to the small proportion of special transactions, the effect of this method is limited. In addition, biryukov et al. [5, 6] proposed a transaction traceability mechanism based on neighbor nodes, which can improve the traceability accuracy by taking neighbor nodes as the judgment basis. However, the scheme needs to continuously send packet to all nodes in Bitcoin network, which may cause serious interference to Bitcoin network.

The network layer anti-anonymity technology has a certain probability to speculate the initial service node IP of the transaction. Gao Feng, Mao Hong-liang and others [3] have achieved the anti-anonymity traceability accuracy with a recall rate of 60% and an accuracy rate of 35.3%. The traceability and positioning from the service node IP to the end-user IP needs to be combined with the operator’s traffic analysis technology and IP positioning data.

Transaction layer anti-anonymity technology refers to finding the correlation between different Bitcoin addresses by analyzing transaction records in Bitcoin ledger, so as to infer the transaction behavior law and capital flow of the transaction address. Liao et al. [7] analyzed the blackmail process of the blackmail software crypto locker by analyzing the Bitcoin ledger data, found multiple Bitcoin addresses belonging to blackmail organizations, and identified a large number of Bitcoin ransom transactions. Meiklejohn et al. [8] used heuristic cluster analysis technology to identify multiple Bitcoin addresses belonging to the Silk Road website. Guo Wen-sheng et al. [9] studied how to realize the division of Bitcoin entities with different types of characteristics through machine learning of Bitcoin ledger data.

Transaction layer anti-anonymity technology can analyze and speculate the characteristics of the trading behavior on the chain of a specific wallet address. Combined with the anti-anonymity label information of the exchange, mining pool and other platform institutions, it can speculate the ownership of some wallet addresses, but it is difficult to determine the user's social identity information. In reality, many Bitcoin hacking incidents generally analyze the transaction data of Bitcoin ledger, track the exchange into which Bitcoin is transferred, and coordinate the exchange to provide user information of the Bitcoin addresses.

In recent years, the research on Bitcoin anti-anonymity technology by integrating data on and off the chain has gradually become a research hotspot. Husam et al. [10] found that Tor Network anonymous services and users by integrating online social network data and Bitcoin ledger data.

5 Behavior Vectors Mapping and Aligning Model

Due to the anonymity of Bitcoin transaction address and trading process, and the poor readability of Bitcoin ledger data, most centralized institutions or platforms, such as exchange and mixed service, will synchronously record the user identity information and behavior information corresponding to Bitcoin ledger data. The above data is called social data off chain. Although it does not contain Bitcoin address, making full use of this data can realize the positioning and anti-anonymity of transaction behavior of Bitcoin ledger data.

We define social behavior vector S including five dimensions: [time, value, scene, name and account]. Time is the time when user receives social data, value is the number of Bitcoin in social data, scene is the transaction scene describing in social information, name is the platform name, and account is the user's social account. If only time and value are considered, and the transaction scene, platform name are missing or ignored, the accuracy of anti-anonymity will be affected in some complex cases.

Like social behavior vector, we define transaction behavior vector E including seven dimensions: [time, value, scene, input label, output label, input address, output address]. Time is the transaction time recorded in the Bitcoin ledger, value is the number of Bitcoin in transaction output, scene is the transaction scene inferred through graph structure analysis, input label is the clustering label of the transaction input address, output label is the clustering label of the transaction output address (non change address), input address is the transaction input address and output address is the transaction output address (non change address). If transaction behavior vector E and social behavior vector S satisfy the following conditions:

- ① Difference between $S.time$ and $E.time$ is small, that is, the social time is close to the Bitcoin ledger transaction time, such as less than 10 min;
- ② $S.Value$ is equal to $E.value$, that is, the transaction values on and off the chain are consistent;
- ③ $S.Scene$ is equal to $E.scene$, that is, the trading scenarios on and off the chain are consistent;
- ④ For deposit transaction, $S.name$ is equal to $E.output$ label, that is, the name of the platform name is consistent with the address clustering label on the chain.

Then, user’s social account S.account corresponding to Bitcoin transaction address E.output address can be considered. Because user’s social account is more unique and social than the IP and user behavior portrait, and can better reflect user’s social identity information.

6 Experiment and Result Analysis

In order to research and prove the alignment model of behavior vector mapping on and off the chain, the anti anonymity of Bitcoin transaction can be realized more accurately. We conducted an experimental test on the charging transaction of a platform. The experimental process is as follows:

- ① Recharge the two deposit wallet addresses assigned by the exchange, then receive 26 social messages sent by the exchange through two social accounts. 26 social messages correspond to 26 social behavior vectors, including 11 social behavior vectors belonging to social account A and 15 social behavior vectors belonging social account B. The sample data of social behavior vector after anonymized is as follows: [‘2020–05-12 14:24’, ‘0.010 *’, deposit, * exchange, ‘account’]
- ② Determine the time window of Bitcoin ledger data. In this experiment, the start time of Bitcoin ledger data is greater than or equal to the social behavior vector’s time minus 20 min, and the end time is less than or equal to the social behavior vector time plus 10 min.
- ③ Extract time and value fields in each social behavior vector, match with the output value of Bitcoin ledger transaction output in the time window, choose Bitcoin ledger transactions output with equal value.
- ④ Analyze the graph structure of the transaction, and choose transaction whose transaction scene is the same as the social behavior vector’s scene.
- ⑤ For the transaction output address, choose address whose cluster label is consistent with the exchange’s name in the social behavior vector.

The experimental results are shown in the following table (see Table 1):

Table 1. Anti-anonymity experimental results of Bitcoin transaction

Social account	Social behavior vectors	Matched social behavior vector	Matched address	Deposit address
A	11	11	1	Yes
B	15	15	1	Yes

Eleven social behavior vectors of social account A are respectively aligned with eleven C2C deposit transaction behavior vectors, and these Bitcoin transaction behavior vectors belong to one Bitcoin address, which is also the deposit address opened by the exchange for user A. Fifteen social behavior vectors of social account B are respectively

aligned with fifteen B2C deposit transaction behavior vectors, and these Bitcoin transaction behavior vectors belong to one Bitcoin address, which is also the deposit address opened by the exchange for user B.

7 Conclusion

The anti-anonymity technology of Bitcoin transaction based on behavior vector mapping and aligning model proposed in this paper, realizes the fusion analysis of data on and off the chain. Compared with the traditional anti-anonymity technology, it has stronger practical effect. At the same time, the anti-anonymity technology proposed in this paper is also applicable to the anti-anonymity of other virtual currencies, such as Ethereum Coin and Tether USD.

References

1. Nakamoto S.: Bitcoin: A peer-to-peer electronic cash system [EB/OL]. <https://www.Bitcoin.org/Bitcoin.pdf,last>. Accessed 18 May 2022
2. Hong-liang, M.A., Zhen, W.U., Min, H.E., Ji-qiang, T.A., Meng, S.H.: Heuristic approaches based clustering of bitcoin addresses. *J. Beijing Univ. Posts Telecom* **41**(2), 27–31 (2018)
3. Gao, F., Mao, H.L., Wu, Z., Shen, M., Zhu, L., Li, Y.D.: Lightweight transaction tracing technology for bitcoin. *Chin. J. Comput.* **41**(5), 899–1004 (2018)
4. Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in Bitcoin using P2P network traffic. In: Christin, N., Safavi-Naini, R. (eds.) *FC 2014*. LNCS, vol. 8437, pp. 469–485. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_30
5. Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in Bitcoin P2P network. In: *Proceedings of the 21st ACM Conference on Computer and Communications Security*, pp. 15–29. New York, USA (2014)
6. Ivan, P.: *Deanonymisation Techniques for Tor and Bitcoin* [Ph.D. dissertation]. University of Luxemb-bourg, Luxemb bourg (2015)
7. Liao, K., Zhao, Z., Doupe, A., et al.: Behind closed doors measurement and analysis of crypto locker ransoms in Bitcoin. In: *Proceedings of the Symposium on Electronic Crime Research*. Toronto, Canada, pp. 1–13 (2016)
8. Meiklejohn, S., Pomarole, M., Jordan, G., et al.: A fistful of Bitcoins: characterizing payments among men with no names. In: *Proceedings of the Conference on Internet Measurement*, pp. 127–140. Barcelona, Spain (2013)
9. Guo, W., Yang, X., Feng, Z., Zhang, L., Yang, J.: Research of de-anonymizing method based on machine learning for bitcoin. *Comput. Eng.* **47**(12), 47–53 (2021)
10. Jawaheri, H.A., Sabah, M.A., Boshmaf, Y., et al.: When a small leak sinks a great ship: deanonymizing tor hidden service users through Bitcoin transactions analysis[EB/OL], <https://arxiv.org/pdf/1801.07501.pdf,last>. Accessed 18 May 2022

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Information Security



Improving Deepfake Video Detection with Comprehensive Self-consistency Learning

Heng Bao¹, Lirui Deng², Jiazhi Guan², Liang Zhang^{3(✉)}, and Xunxun Chen³

¹ School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

baoheng@iie.ac.cn

² Department of Computer Science and Technology, Tsinghua University, Beijing, China

{dlr18, guanjz20}@mails.tsinghua.edu.cn

³ CNCERT/CC, Beijing, China

zl@isc.org.cn

Abstract. Deepfake videos created by generative-base models have become a serious societal problem recently as been hardly distinguishable by human eyes, which has aroused a lot of academic attention. Previous researches have made effort to address this problem by various schemes to extract visual artifacts of non-pristine frames or discrepancy between real and fake videos, where the patch-based approaches are shown to be promising but mostly used in frame-level prediction. In this paper, we propose a method that leverages comprehensive consistency learning in both spatial and temporal relation with patch-based feature extraction. Extensive experiments on multiple datasets demonstrate the effectiveness and robustness of our approach by combines all consistency cue together.

Keywords: Deepfake detection · Digital forensics · Video classification

1 Introduction

“Seeing is believing” is hardly true in present days with the prosperity of computer science and information technology, especially the massively emerging applications of artificial intelligence. Although image and video forgery is never a new topic since the beginning of photography, open source applications represented by Deepfakes [7] and others have brought this problem into a whole new level. Face manipulation in visual content has become a effortless task with the help of deep learning based generative models like variational autoencoders (VAEs) [16] and generative adversarial networks (GANs) [12], that anyone can produce fake videos with false identity or manipulated expressions and movements (known as “Deepfake Videos”) in several minutes without expert knowledge. Some of them have already been found to create malicious videos that

violate citizen privacy or attack public figures like fake pornography and blackmailing, which may easily lead to catastrophic results with today’s mass media and social networks. The detection of deepfake videos has become an hot and urgent issue.

Various methods have been proposed in recent years by academic community to effectively recognize this particular type of forged images and videos. Since the majority forgery methods share a common image stitching pipeline including face detection, warping and blending, early researches address this task by detecting suspicious artifacts left in the stitching process within frame-level, such as face warping artifacts [20] and blending boundaries [18]. To yield a result for whole video clip from frame-level prediction, they usually cascade frame-level model with a merge module, or sometimes simply use weighted average. But ignoring the dependency among consecutive frames tends to produce sub-optimal combination. Frequency-based approaches [9,25] have also been included to fully utilized temporal relation. Self-consistency is another crucial concept in image forensic [14,35], where patch-based and feature-map based method have all shown promising results. Although the detection accuracy on datasets has improved significantly with different approaches presented, forgery techniques are also evolving on reducing these artifacts, which forms an ever-changing arms race.

In this work, we aim to catch both the intra-frame discrepancy during image stitched and the inherent flaws of inter-frame disalignment for more effective and robust deepfake detection. Our contributions can be summarized as follows:

- We propose a comprehensive self-consistency learning(CSCL) model to explore the intrinsic discernible evidence between pristine and deepfake videos with both spatial and temporal consistency learning.
- To achieve more effective and robust deepfake detection, we also proposed C^3Loss , namely comprehensive consistency coordination loss, which tackles the inherent defects within deepfake producing pipeline as been created frame-by-frame without sequential knowledge.
- Experiments conducted on multiple datasets demonstrate the effectiveness and robustness of our approach. Especially, best performance is reported in cross-dataset and low quality tests.

2 Related Work

Frame-Level Detection Methods. The emerging of deepfake videos on the internet raise a lot of concern to both industry and government in the past few years. Early researches [1,26] tend to address this problem by a simple classification model with a well-designed backbone. And some [20] simulated the generation process of deep forgery to better obtain artifact of fake video pipeline. Not only in academic society, a one-million bounty real-world deepfake detection competition was held by Facebook with the concern of its endangerment of social media to encourage optimal deepfake detection methods being proposed. Plenty of classification model was proposed and achieve really amazing results

beyond expectation. The winner of this competition [27] adopts the state-of-the-art image classification backbone efficientNet [28] as the main component of his model, and a novel data argumentation strategy contributes a lot to his final ranking. The runner-up of this competition fulfilled their method afterward [34], which treat the deepfake detection task as a fine-grained classification task and explicitly refine attention maps by regional independence loss. Merging with texture features extracted at the front-end layer, their model achieve state-of-the art results in several datasets. Attention map prediction scheme is also considered by [6]. In their work, forged area is predicted in both learning-base and dictionary learning ways, binary classification and attention map regression tasks are trained using a multi-task loss function.

The above mentioned detection methods are all concentrate on the RGB-domain of deepfakes, and there are some other works try to explore fake clues inherent in the frequency domain of deepfake images. Discrete cosine transform (DCT) [2] is adopted in [25], frequency layout of image is fully handled in both global and local views. In combine with learnable frequency-aware component, nonaligned infomation can be reliably detected at frame-level. Frank *et al.* [9] also leverage DCT in detection, and analysis which part makes synthetical deepfake image detectable. Their results suggest that up-sampling blocks left unique fingerprint, but those frequency clues are not robust to perturbation.

Besides, some other approaches tried to inspect artifacts from the side-view. FakeSpotter [31] do not directly use the features extracted by backbone network, but regard the neuron behaviors as the basis of discrimination, which is aimed to achieve more robust detection. To better leveraging the time factor into consideration in video-level authentication, spotting bio-metrics clues like eye blinking [19,32] and head posing sequential [32] is the first and most natural insight. DeepRhythm [24] exposes deepfake counterfeits by monitoring the heartbeat rhythms associated with minuscule periodic changes of skin color due to blood pumping through the face.

Video-level Detection Methods. Most video-level methods regard video as set of independent frames, and simply take the average confidence score of frames as the basis of judging the authenticity of video. Those methods actually follow the frame-level perspective, and neglect the interconnection between successive frames.

Güera *et al.* [13] adopt a natural way to leverage both advantages of convolutional neural network (CNN) and recurrent neural network (RNN) by using CNN for per-frame feature extraction, and RNN for temporal inconsistencies exploration. But inter-frame inconsistency modeling is not well considered in their approach. Tariq *et al.* [29] consider the artifacts introduced by the non-consecutive frames, and developed a convolutional LSTM-base residual network to achieve temporal feature learning. Basic features of the human body like eye blinking and head pose moving are utilized in [19] and [32] to distinguish the real from fake. In [23], the authors leverage the relationship between visual and audio patterns extracted from the same video to determine whether it has been modified.

Video-level detection gathers more information and, in general, should deliver better performance. But strangely, video-level evaluation results in terms of ACC and AUC are somehow lower than those at the frame-level. Zi *et al.* [37] propose two models by stacking ADD block. In their experiments, ADDNet-3D report much lower detection accuracy than ADDNet-2D, about 10 percents gap at a challenging dataset. Ganiyusufoglu *et al.* [11] adopt the state-of-the art structures used in action recognition task, and evaluate their performance in deepfake detection.

Benchmark Datasets. Several comprehensive deepfake datasets were published in recent years which greatly promotes the performance of deepfake detection methods. One of the most popular dataset is FaceForensics++ (FF++) [26]. It contains two graphic based approaches, namely Face2Face [30] and Faceswap [8], and two learning based methods include Deepfakes and Natural Textures [15]. Both face swap and face reenactment are covered. Celeb-DF [21] is one of the most challenging dataset in deepfake detection task with clear identity label and pixel level annotation. During the deepfake generation stage, they scrutinizes carefully about several problems during fake video generation, including color mismatch, inaccurate face masks and video temporal flickering. With more attention drawn to this research topic, some new and better annotated datasets been proposed with more specific purpose recently like WildDeepfake [37] for real-world challenge and OpenForensic [17] for multiple face scenario.

3 Approach

Given an input video with certain human activity, our goal is to detect if the identity is replaced or facial expression of character is manipulated. We propose CSCL network as shown in Fig. 1 to improve the robustness and generalization ability of deepfake-style forgery video detector with the help of self-consistency by measuring the comprehensive spatial and temporal discrepancy within the image stream.

To be more precise, our method mainly exploit a comprehensive consistency which tackles the substantial drawback of deepfake videos producing pipeline:

- **Intra-frame: Spatial consistency.** Intra-frame consistency in deepfake video are mostly provided by blending algorithm like Gaussian blur or Poisson fusion, which has been proved to be distinguishable.
- **Inter-frame: Temporal consistency.** Common generative models with frame-by-frame swapping process can not guarantee a smooth temporal momentum, while most of the former consistency learning methods only focus on single manipulated frame and tend to be overfit in one subset of manipulation.
- **Comprehensive consistency coordination.** Extra blur and filtering can conceal the intra-frame discrepancy, and inter-frame consistency may also be diminished by adaptive average blending. Unlike previous work, we utilize inter- and intra-frame consistency coordination for more robust deepfake video detection.

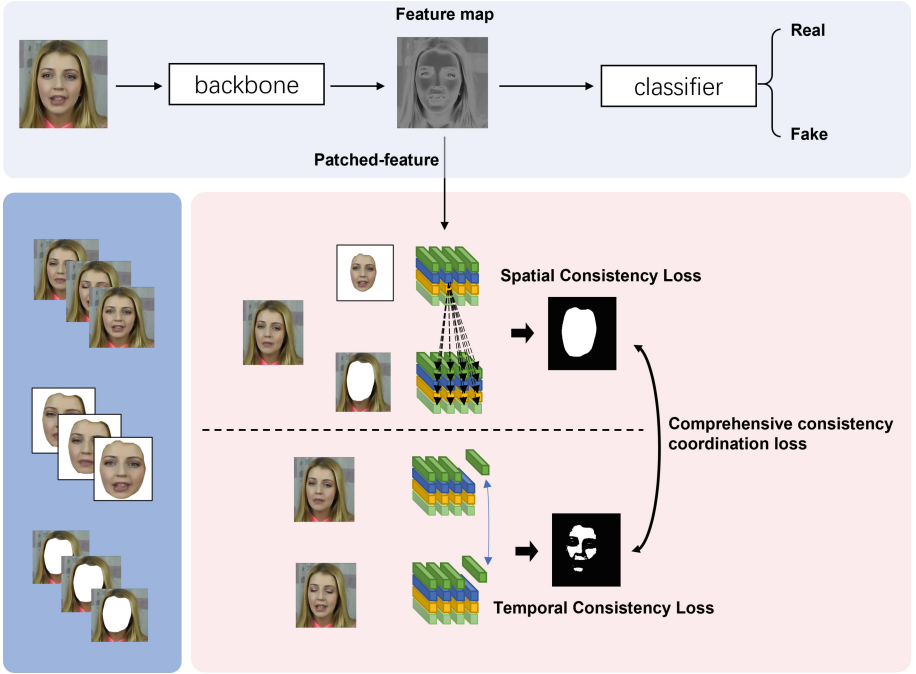


Fig. 1. Framework of CSCL network

3.1 Problem Formulation

We formulate the video-level deepfake detection task at beginning. Dataset $D = \{X_i, L_i\}_{i=1}^N$ consists of n pairs of video-clip and its label with **fake** or **real** denoted as $L_i = \{0, 1\}$. Video clip can be seen as multiple consecutive frames $X_v = \{x_t\}_{t=1}^{T_v}$, where $x_t \in \mathbb{R}^{C \times H \times W}$ is the t -th frame of video X_v , and the total number of frames is denoted as T_v . All the frames in one specific video X_v are deemed as manipulated if X_v is labeled with **fake**, vice versa. The goal of deepfake detection is to learn a model Φ , which takes all consecutive frames of one video, and give a clear judgment of the authenticity, formulated as $\Phi(X_v) \in \{\text{fake}, \text{real}\}$.

3.2 Design of Model

Spatial Consistency of Contexts vs. Faces. Computing similarity scores among images patched for inconsistency has already been proved effective in image forensic researches [33, 35, 36]. Without loss of generality, we first obtain feature f_t of image x_t from backbone model \mathbf{G} of size $H' \times W' \times C'$ where H' and W' and patch numbers along columns and rows.

$$f_t = \mathbf{G}(x_t)_{t=1}^{T_v} \in \mathbb{R}^{C' \times H' \times W'} \quad (1)$$

For each frame x_t we follow the [35] to calculate the 4D consistency map $\hat{S}M$ with:

$$\begin{aligned} \hat{S}M_{h,w,h',w'} &= d(f_t^{h,w}, f_t^{h',w'}) \\ &= 1 - \cos(f_t^{h,w}, f_t^{h',w'}) \end{aligned} \quad (2)$$

While each frame’s mask have only two possible status : manipulated or not, for patch P located in face area denoted as P_f , else in context as P_c , and $\psi(P_f) = 1$ else $\psi(P_c) = 0$, the ground truth:

$$SM_{P_i,P_j} = \psi(P_i) \oplus \psi(P_j) \quad (3)$$

and the spatial consistency loss:

$$L_{SC} = |SM - \hat{S}M| \quad (4)$$

Temporal Consistency of Consecutive Frames. In order to catch inconsistency between successive frames, we further extend the attention to temporal consistency learning. As we have obtained the patch-base feature f_t from x_t , we consider the relation between f_t and f_{t-1} . For each path $P_{h,w}$ at timestamp t , we have a 2D consistency map:

$$\begin{aligned} \hat{T}M_{P_t^{h,w} P_{t-1}^{h,w}} &= d(f_t^{h,w}, f_{t-1}^{h,w}) \\ &= 1 - \cos(f_t^{h,w}, f_{t-1}^{h,w}) \end{aligned} \quad (5)$$

considering the momentum between t and $t - 1$, we calculate temporal consistency loss:

$$L_{TC} = \sum_t |T\hat{M}_t - \frac{\sum_{h,w} T\hat{M}_{t,h,w}}{HW}| \quad (6)$$

Coordinating Temporal and Spatial Consistency. It’s not hard to imaging that no matter in pristine or deepfake video people’s face will be moving most of the time, either talking or acting expressions. Otherwise the there’s no need to forge this static video which conveys no more information than just a photo. Only measuring the discrepancy of distance between consecutive face and context would yield lots of false alarm. Therefore we propose a comprehensive consistency coordination loss for adaptive learning by monitoring the relation between temporal and spatial consistency. Now we have final Loss function:

$$L = L_{real/fake} + \lambda L_{SC} + \beta L_{TC} + (1 - \beta) L_{CCC} \quad (7)$$

4 Experiment Results

Implementation Details. We modify Xception [4] as the backbones and their parameters are initialized by Xception pre-trained on ImageNet. We train our model using Adam optimizer with initial learning rate 1e-4 and weight decay 1e-7. Train epoch size is set to 2000, batch size is set to 32, and if validation loss is not getting better in 5 epochs, learning rate is decayed by factor 0.3, so that model can converge after several learning rate decays.

4.1 In-Dataset Evaluation on FF++

FF++ is one of the most popular dataset for evaluating deepfake detection methods. It contains 1000 real videos collected from internet, and 4000 fake videos generated by four kinds of deepfake techniques. More over, FF++ provides 3 different qualities of videos, we use the high quality (c23) and low quality (c40) versions in this section. The raw quality videos are not considered because they are not very common on the internet. We use the same split as [26], both real and fake video is split into train, validation and test set according to the ratio of 72:14:14. But it is noticed that number of real videos is much smaller than fake videos. So, we over-sample real videos to balance the classes when training. At test stage, one video could contain several clips in FF++, we extract as much clips as we can from one video (interval is set to 16, no overlap), and take the average score of all clips as confidence score of the video. The test results are listed in Table 1.

Table 1. In-dataset Performance (ACC %) on four types of deepfake in FF++. DF: DeepFakes, F2F: Face2Face, FS: FaceSwap, NT: NeuralTextures. The best result is shown in bold text, and the second-best is underlined.

	Methods	DF	F2F	FS	NT
Frame Level	LD-CNN [10]	75.00	56.00	51.00	62.00
	Constrained Conv [5]	87.00	82.00	74.00	74.00
	CustomPooling CNN [3]	80.00	62.00	59.00	59.00
	MesoNet [1]	<u>90.00</u>	<u>83.00</u>	<u>83.00</u>	83.00
	Xception [4]	96.01	93.29	96.71	<u>79.14</u>
Video Level	PCL [35]	96.87	94.93	<u>98.44</u>	99.58
	PD [33]	<u>97.53</u>	<u>96.57</u>	95.01	92.55
	ours	100.00	99.84	99.21	<u>99.37</u>

4.2 Cross-Dataset Evaluation on Celeb-DF

The poor Generalization ability of deepfake detection is still a thorny problem, even the state-of-the-art methods suffer from drastically performance degradation when test on deepfakes generated by unseen techniques. Our method tries to formulate deepfake detection from a discrepancy discovering aspect, and achieves the best cross-dataset performance, as the results listed in Table 2. The test model is trained on FF++ low quality, follow the setting of [22] for fair comparison. It is noticed that many methods report around 100% AUC on train set, but fail to transfer to the different dataset. Our model achieve the best cross-dataset test performance, while keep the best test result on train set.

Table 2. Cross-dataset Performance (AUC%) on Celeb-DF. The best result is shown in bold text, and the second-best is underlined.

Methods	FF++	Celeb-DF
MesoNet-Inception [1]	83.00	53.60
FWA [20]	80.10	56.90
Xception-raw [4]	99.70	48.20
Xception-c23 [4]	99.70	65.30
Xception-c40 [4]	95.50	65.50
DSP-FWA [20]	93.00	64.60
Two-Branch [22]	93.18	73.41
PCL [35]	99.79	72.44
Patch-Diffusion [33]	99.85	<u>74.27</u>
ours	99.85	77.73

4.3 Ablation Study

This section analyzes the effectiveness of our proposed CSCL module. CSCL consist of three parts in total: the spatial consistency, temporal consistency and comprehensive consistency coordination. To further validate whether each part of comprehensive consistency can improve the generalizability, we conduct an ablation study by comparing our methods with the following variant. (1)Xception [4]: the baseline approach without using any consistency cue. (2)Xception w/ sc: we follow the setting of [35] with only spatial patch consistency loss. (3)Xception w/ tc: we use only temporal consistency loss upon baseling. (4)Ours full CSCL model with both spatial and temporal consistency, plus consistency coordination loss. Results are listed in Table 3.

Table 3. Ablation Performance in FF++. The best result is shown in bold text.

	Methods	AUC(HQ)	AUC(LQ)
Base Line	PD [33]	99.85	94.43
	PCL [35]	99.79	96.38
Ablation	Xception+SC	98.46	98.01
	Xception+TC	95.13	94.93
	Xception+SC+TC	99.46	98.16
	CSCL(SC+TC+CCC)	99.85	98.21

5 Summary

In this paper, we try to address the problem, deepfake detection, from the view of comprehensive self-consistency learning. More specifically, we propose a CSCL

model with spatial-temporal consistency learning to explicitly formulate the inherent flaws of intra- and inter-frame disalignment in deepfakes. To achieve more effective and robust deepfake detection, we also proposed C^3Loss , namely comprehensive consistency coordination loss, which tackles the inevitable artifact within deepfake producing pipeline. Extensive experiments demonstrate the superior performance of our method in deepfake detection, especially in more realistic tests like cross-dataset and low quality setting.

References

1. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: MesoNet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7. IEEE (2018)
2. Ahmed, N., Natarajan, T., Rao, K.: Discrete cosine transform. *IEEE Trans. Comput.* **23**(01), 90–93 (1974)
3. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10 (2016)
4. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
5. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 159–164 (2017)
6. Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5781–5790 (2020)
7. Deepfake github. <https://github.com/deepfakes/faceswap>
8. Faceswap github. <https://github.com/MarekKowalski/FaceSwap/>
9. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: International Conference on Machine Learning, pp. 3247–3258. PMLR (2020)
10. Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur. (TIFS)* **7**, 868–882 (2012)
11. Ganiyusufoglu, I., Ngõ, L.M., Savov, N., Karaoglu, S., Gevers, T.: Spatio-temporal features for generalized detection of deepfake videos. *arXiv preprint arXiv:2010.11844* (2020)
12. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in neural information processing systems, vol. 27 (2014)
13. Güera, D., Delp, E.J.: Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2018)
14. Han, X., Morariu, V., Larry Davis, P.I., et al.: Two-stream neural networks for tampered face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 19–27 (2017)
15. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)

16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
17. Le, T.N., Nguyen, H.H., Yamagishi, J., Echizen, I.: Openforensics: large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10117–10127 (2021)
18. Li, L., et al.: Face x-ray for more general face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5001–5010 (2020)
19. Li, Y., Chang, M.C., Lyu, S.: In Ictu oculi: exposing AI created fake videos by detecting eye blinking. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7. IEEE (2018)
20. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. arXiv preprint [arXiv:1811.00656](https://arxiv.org/abs/1811.00656) (2018)
21. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-DF: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3207–3216 (2020)
22. Masi, I., Killekar, A., Mascarenhas, R.M., Gurudatt, S.P., AbdAlmageed, W.: Two-branch recurrent network for isolating deepfakes in videos. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 667–684. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_39
23. Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., Manocha, D.: Emotions don't lie: a deepfake detection method using audio-visual affective cues. arXiv preprint [arXiv:2003.06711](https://arxiv.org/abs/2003.06711) (2020)
24. Qi, H., et al.: Deeprhythm: exposing deepfakes with attentional visual heartbeat rhythms. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 4318–4327 (2020)
25. Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J.: Thinking in frequency: face forgery detection by mining frequency-aware clues. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12357, pp. 86–103. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58610-2_6
26. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: learning to detect manipulated facial images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1–11 (2019)
27. Selim, S.: Deepfake detection (DFDC) solution by selim seferbekov. https://github.com/selimsef/dfdc_deepfake_challenge
28. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
29. Tariq, S., Lee, S., Woo, S.S.: A convolutional LSTM based residual network for deepfake video detection. arXiv preprint [arXiv:2009.07480](https://arxiv.org/abs/2009.07480) (2020)
30. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: real-time face capture and reenactment of RGB videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2387–2395 (2016)
31. Wang, R., et al.: Fakespotter: a simple yet robust baseline for spotting AI-synthesized fake faces. arXiv preprint [arXiv:1909.06122](https://arxiv.org/abs/1909.06122) (2019)
32. Yang, X., Li, Y., Lyu, S.: Exposing deep fakes using inconsistent head poses. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8261–8265. IEEE (2019)

33. Zhang, B., Li, S., Feng, G., Qian, Z., Zhang, X.: Patch Diffusion: a general module for face manipulation detection. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence (2022)
34. Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., Yu, N.: Multi-attentional deepfake detection. arXiv preprint [arXiv:2103.02406](https://arxiv.org/abs/2103.02406) (2021)
35. Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., Xia, W.: Learning self-consistency for deepfake detection. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15003–15013 (2021). <https://doi.org/10.1109/ICCV48922.2021.01475>
36. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Neural networks for tampered face detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1831–1839. IEEE (2017)
37. Zi, B., Chang, M., Chen, J., Ma, X., Jiang, Y.G.: WildDeepFake: a challenging real-world dataset for deepfake detection. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2382–2390 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Research on Information Security Asset Value Assessment Methodology

Xueqin Yang¹, Peng Yang², and Honggang Lin¹(✉)

¹ School of Cyberspace Security, Chengdu University of Information Technology,
Sichuan 610225, Chengdu, China

1844108560@qq.com

² National Computer Network Emergency Technology Processing Coordination Center,
Beijing 10009, China

Abstract. In response to the fact that traditional asset value assessment methods are subjective and cannot distinguish the value of different assets carrying the same type of business, a comprehensive assessment method that takes into account the importance of the business carried by the assets is proposed. In this paper, four factors affecting business importance are selected as evaluation indicators, and the CRITIC objective assignment method is used to obtain the weights of each evaluation indicator, calculate the importance of the business carried by the asset, and then calculate the asset value using the multiplication method with the assigned values of the asset in terms of confidentiality (C), integrity (I) and availability (A). The results of the case validation show that the calculation results of assessing the asset value by combining business importance are consistent with the actual value of the asset, and the comparison results with the traditional method show that the proposed method is more objective and reasonable in assessing the asset value.

Keywords: Asset value assessment · Business importance · CRITIC objective empowerment method · Multiplication method

1 Introduction

As government departments, financial institutions, enterprises and institutions, and commercial organizations rely on information systems, information security issues have received widespread attention and importance. Using risk assessment to analyze the security risks in information systems and propose targeted corrective measures is an effective means to solve information security problems. Among them, identifying assets and assessing their value is the primary task of information security risk assessment, and the current calculation of asset value is mainly to be achieved based on confidentiality (C), integrity (I) and availability (A) [1]. Tang [2] proposes an objective assignment method to assign weights to evaluation indicators to make the calculated importance values more objective, but this weighting method only considers the dispersion of data and does not consider the correlation between indicators. In the literature [3], it is proposed that since quantifying the security level of assets in terms of confidentiality, integrity, and

availability is prone to subjectivity, the importance of the business carried by the assets is considered as a factor to reduce the subjective influence, and then using weighting and other methods to synthesize the value of the assets, but the literature does not give a specific implementation algorithm. Xiang Hong [4] identifies assets based on business and uses the AHP method to assign values to assets. The AHP method is effective in reducing the drawbacks of being completely subjective, but the method requires multiple experts with rich experience to give a reliable judgment matrix and also involves calculations such as consistency tests, which increases the complexity of the calculations. Zhou Jing-Xian [5] uses the rough set approach to calculate the value of each asset by assigning weights to four factors of CIA that determine the value of the asset and the importance of the business undertaken, and since the importance to the business is judged by human, then once the decision makers differ, it will result in a situation where the same asset has different values. In order to solve the above problems, it is necessary to propose to calculate the asset value by combining the importance of the business carried by the quantified assets.

In this paper, we propose a method to evaluate the asset value by using the importance of the business carried by the asset together with the four factors of confidentiality, integrity and availability. The method selects the evaluation indexes that can reflect the importance of the business carried by the asset and calculates the business importance value by combining the CRITIC weighting method, and then uses the multiplication method for the four influencing factors to obtain the asset value, which can reduce the subjective influence of the traditional method when considering CIA and distinguish the value of assets that belong to different organizations but carry the same type of business and assets that carry different types of business under the same organization.

2 Asset Valuation Method

The value of an asset is determined by the level of assignment of the three security attributes of confidentiality, integrity and availability, as well as the importance of the business undertaken by the asset. The realization of a complete business requires the involvement of multiple assets, and the more significant the business is, the more important its associated assets are. Based on this, this paper proposes an intuitive asset valuation model to analyze the value of assets.

2.1 Asset Valuation Model

The asset value assessment model proposed in this paper is depicted in Fig. 1. For information assets, their value is mainly reflected in four indicators: confidentiality, integrity, availability and the higher the requirements for these indicators, the higher the asset value. The three security attributes of confidentiality, integrity and availability are classified into five levels: very high, high, medium, low and very low, and the higher the level, the higher the requirement of the asset for this security attribute. The importance of the business carried by the assets is mainly reflected in the business itself and the impact of the assets attached to the business on the organization, so the importance of the business can be evaluated from four aspects: organization ranking, organization level, scope of impact and business category.

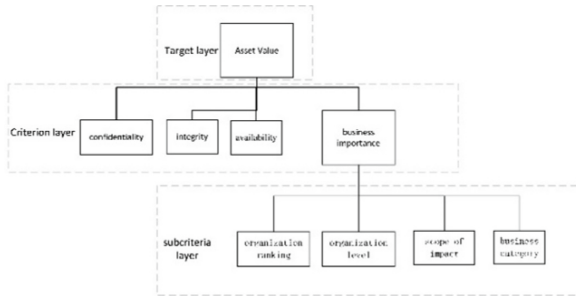


Fig. 1. Asset valuation model

2.2 Business Importance Indicators System Construction

From the evaluation model, it is obvious that the asset value will be affected by the importance of the business. In order to evaluate the asset value more accurately, it is required to select indicators that rely completely on objective data to quantify the importance of the business. In this paper, the business category and influence range indicators proposed by the business itself point out that the more core the business category is, the higher the importance of the business, and the more extensive the influence range is when the business cannot operate normally, the higher the importance of the business. However, considering that the selected indicators do not fully reflect the importance of the business and the indicators proposed from the business itself cannot distinguish the importance of different businesses that belong to the same business category and have the same scope of influence, this paper based on the assets on which the business depends, and proposes two indicators, organization ranking and organization level, to reflect the importance of the business running on them by measuring the importance of the assets. Among them, organization ranking refers to the ranking of the organization to which the asset belongs within the industry. The higher the ranking, the stronger the organization is in the industry and the higher the importance of its subordinate assets; The organization level refers to the category in which the organization to which the asset belongs is classified in that industry. The higher the category level belongs to, the more important the organization is and the more important its subordinate assets are. (If the value of an indicator of the assessment object cannot be determined, we may assign the same default value to the indicator and it is necessary to ensure that the final sum of all indicator weights is 1.)

With regard to the organization ranking, organization level, and influence range indicators, it is necessary to analyze the reports issued by the organization to which the actual assets belong to obtain their values, while the business category indicator can be determined by initially knowing the classification of the business according to the literature [6] and then combining it with the business carried on the actual assets to identify the specific category. The literature roughly classifies businesses into five major categories according to their characteristics (since specific business systems are not mentioned, the information in the table is not complete, and the classification of businesses in the actual assessment work should be based on the actual situation), as shown in Table 1. Because the value of an asset takes into account the importance of the

business it carries, it is likely that the exact same information asset will have a different value to the organization being evaluated because of the different businesses it carries.

Table 1. Example of business classification.

Business category	Business data characteristics	Assignment
Production application business class	Directly linked to core business operations	5
Financial marketing business class	Handling confidential internal operations and classified data	4
Management information business class	Office automation and other management information services	3
Open business class	Direct to external users	2
Other businesses	Ensure the normal operation of the basic system	1

2.3 Business Importance Calculation

Because of the differences in the contribution of each indicator to the importance of business, this paper uses the CRITIC method [7] to assign weights to indicators to calculate the importance of business. As can be seen from the previous subsection, business importance is determined by four indicators: organization ranking, organization level, influence range, and business category, and the CRITIC method which takes into account the conflicting nature of the indicators and the characteristics of the differences in the values taken by the evaluation objects under each indicator is used to calculate the weight of each indicator [8]. For example, if there is a greater conflict between the organizational ranking indicator and other indicators, the greater the difference in the data under that indicator, which means that the indicator contains more information, that is, it has greater weight and contributes more to the importance of the business. Similarly, the weights of other indicators can be obtained from the CRITIC method [9], and the calculation steps are as follows:(In this paper, if we select only one indicator to assess the importance of the business, we only need to do the normalization step of the indicator data in this algorithm).

- 1) In order to eliminate the influence on the evaluation results of different magnitudes, formula (1) was used to reverse the process for the indicators belonging to the smaller value, and formula (2) was used to forward the process for the indicators belonging to the larger value [10]:

$$x_{ij}^* = \frac{\max(x_j) - x_{ij}}{\max(x_j) - \min(x_j)} \tag{1}$$

$$x_{ij}^* = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{2}$$

In the formula: $\max(x_j)$ is the maximum value of the j indicator, $\min(x_j)$ is the minimum value of the j indicator, x_{ij} is the value of evaluation object i under indicator j , x_{ij}^* is the processed value and its value range is $[0,1]$.

- 2) After the data were processed, the standard deviation of each indicator was calculated using formula (3) as an indication of the difference in the values taken by each assessment subject under each indicator:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij}^* - \bar{x}_j)^2} \tag{3}$$

Among them, σ is the standard deviation of the j indicator and is the average of n assessment objects under indicator j .

- 3) Formulas (4) and (5) are used to calculate the magnitude of conflict between indicators:

$$r_{ij} = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2}} \tag{4}$$

$$A_j = \sum_{i=1}^m (1 - r_{ij}) \tag{5}$$

In the formula, r_{ij} denotes the correlation coefficient between indicator i and indicator j , x_{ik} and x_{jk} denote all data under indicator i and indicator j , respectively, and A_j denotes the conflict between indicator j and other indicators.

- 4) From formula (6), the weights of each indicator is w_1, w_2, w_3 and w_4 :

$$w_j = \frac{\sigma_j A_j}{\sum_{j=1}^M \sigma_j A_j} \tag{6}$$

- 5) According to the weight of each indicator and the value of each business object under each indicator, the business importance α can be obtained:

$$\alpha = \sum_{j=1}^M w_j x_{ij}^* \tag{7}$$

2.4 Asset Value Calculation

After obtaining the asset’s assigned level of confidentiality, integrity and availability and the importance of the business it carries, then we use the multiplication method to calculate the asset’s value. The specific calculation steps are as follows:

Set the value of the j asset as d_j , its values in confidentiality, integrity, and availability as c, i , and a , and its business importance as α . The formula of calculating the asset value is as follows:

$$d_j = \sqrt[3]{\alpha * c * i * a} \tag{8}$$

3 Evaluation Examples and Results Analysis

This paper uses the bank assets which are obtained from the extranet as the valuation object, and applies the above calculation method to calculate the value of each asset, and compares and analyzes the results obtained with the traditional method.

3.1 Instance Data

In order to determine the value of the indicators which are selected for the calculation of business importance, the analysis will be performed here in combination with the actual assets. From the literature [11], we know the ranking of organizations to which bank extranet assets belong, and from the literature [12], we can summarize and classify the organizations of bank information system assets into five major levels: state-owned large banks, state-owned commercial banks, regional urban commercial banks, rural banks in each county and district, and private banks. For the bank information system, the impact range of the business on it can be reflected by the impact range of the organization to which the asset belongs, so the impact range can be divided into five categories: global, national, province/municipality/autonomous region, city, and county, and assign values in descending order of range. Combined with the actual evaluation object and according to the literature [13], the categories of services carried by the bank's extranet assets can be classified into five major categories: transaction-type services, customer exchange services, online investment services, information services and other services. The transaction services include money transfers and credit operations performed by individuals or companies, which are the highest level of banking service systems and definitely have access to the bank's internal network. The customer exchange service is the communication of information, documents or files between the customer and the bank [14, 15], and this kind of service is a higher-level service system and has access to the bank's internal network. The online investment service [16] is a service that provides customers to purchase various types of financial products launched by the bank. The information service is to publish information that can be accessed by everyone, and this type of service is the most basic type of business that has no access to the bank's internal network. The other services include various forms of special value-added services, such as life type payment services. The business categories are assigned according to the degree of connection to the bank's internal and the level of the service system, see Table 2. Through the above analysis, the 18 acquired bank extranet assets are organized as shown in Table 3. (The 18 selected assets S_1 - S_{18} are ICBC about ICBC system, China Construction Bank deposit and loan and bank card system, China Construction Bank investment and finance system, Agricultural Bank of China personal service system, Agricultural Bank of China talent recruitment system, Bank of China personal financial system, Bank of China electronic banking system, Bank of JIANGSU personal business system, Chongqing Rural Commercial Bank savings business system, Chengdu Rural Commercial Bank's personal financial service system, Bank of Chongqing's personal business system, TRC Bank's savings business system, Bank of Dongguan's personal business system, NRC Bank's savings business system, Bank of Tangshan's email system, XIAOSHAN Rural Commercial Bank Savings Business System, ZJB's savings business system.)

Table 2. Assignment of business importance indicators.

Level assignment	Bank level	Scope of influence	Business category
5	Large state-owned enterprises	Global	Trading business
4	State-owned commercial banks	National	Customer communication services
3	City commercial banks	Province/Municipality/Autonomy	Online investment services
2	Rural banks	City	Information services
1	Private Banks	County	Other services

Table 3. Asset information form.

Asset number	Organization ranking	Organization category	Influence scope	Business type
S1	1	Large state-owned enterprises	Global	Trading business
S2	1	Large state-owned enterprises	Global	Information services
S3	2	Large state-owned enterprises	Global	Trading business
S4	2	Large state-owned enterprises	Global	Online investment services
S5	3	Large state-owned enterprises	Global	Trading business
S6	3	Large state-owned enterprises	Global	Information services
S7	4	Large state-owned enterprises	Global	Trading business
S8	4	Large state-owned enterprises	Global	Customer communication services

(continued)

Table 3. (continued)

Asset number	Organization ranking	Organization category	Influence scope	Business type
S9	18	City commercial banks	Province	Trading Business
S10	22	Rural banks	County	Trading business
S11	36	Rural banks	County	Trading business
S12	44	City commercial banks	Municipality	Trading business
S13	53	Rural banks	County	Trading business
S14	62	City commercial banks	City	Trading business
S15	70	Rural banks	County	Trading business
S16	88	City commercial banks	City	Other services
S17	95	Rural banks	County	Trading Business
S18	100	Rural banks	County	Trading Business

3.2 Business Importance Calculation

We use organization ranking (Index 1), organization level (Index 2), scope of influence (Index 3), and business type (Index 4) as four indicators to assess the importance of the business carried on the bank’s assets. The four indicators are quantified in Table 2, and the results are shown in Table 4.

Table 4. Quantification of business importance indicators.

Asset Number	Index 1	Index 2	Index 3	Index 4	Asset Number	Index 1	Index 2	Index 3	Index 4
S1	1	5	5	5	S10	22	2	1	5
S2	1	5	5	2	S11	36	2	1	5
S3	2	5	5	5	S12	44	3	3	5
S4	2	5	5	3	S13	53	2	1	5
S5	3	5	5	5	S14	62	3	2	5
S6	3	5	5	2	S15	70	2	1	5
S7	4	5	5	5	S16	88	3	2	1
S8	4	5	5	4	S17	95	2	1	5
S9	18	3	3	5	S18	100	2	1	5

We use formula (1) and formula (2) to forward or reverse the values of assets under the above four indicators. For assets, the smaller the value under the organization ranking indicator, the better, while the larger the value under the three indicators of organization level, impact area, and business category, the better. From Table 4, it can be seen that asset S3 takes the value x_{31} of 2 under the organization ranking indicator, the data under this indicator has a maximum value of 100 and a minimum value of 1. Replacing into formula (1), the value of x_{31} after reverse processing can be obtained as $x_{31}^* = \frac{100-2}{100-1} = 0.9899$. The value x_{32} of asset S_3 under the business category indicator is 5, and the maximum value of data under this indicator is 5 and the minimum value is 2. Replacing into formula (2), we can get $x_{32}^* = \frac{5-2}{5-2} = 1$. Similarly, the values x_{33} and x_{34} of S_3 under the influence range and business category indicators are $x_{33}^* = 1$ and $x_{34}^* = 1$ respectively after processing by formula (2). Similarly, the values of other assets under the four indicators are processed similarly.

After the above processing, the mean value of each indicator can be found as 0.669, 0.519, 0.528, and 0.819 in order. We then substituted the 18 data under the organization ranking index and the mean value of the index into formula (3) that we can find the standard deviation of the index is 0.362, and the standard deviation of the other indexes is similar to this, and the results are shown in Table 5. From formula (4) and formula (5), we can find the magnitude of conflict between each indicator and other indicators as 1.457, 1.558, 1.486, and 3.735.

From formula (6), we can obtain the weight w_1 of the organizational ranking indicator:

$$w_1 = \frac{0.362 * 1.457}{0.362 * 1.457 + 0.460 * 1.558 + 0.461 * 1.486 + 0.330 * 3.735} = 0.167$$

Similarly, the weight w_2 of the organization level indicator is found to be 0.227, the weight w_3 of the influence range indicator is 0.217, and the weight w_4 of the business category indicator is 0.389.

Table 5. CRITIC method to calculate the weighting process.

Index	Standard deviation	Conflicting indicators	Weights
Index1	0.362	1.457	0.167
Index2	0.460	1.558	0.277
Index3	0.461	1.486	0.217
Index4	0.330	3.735	0.389

From Table 4, the values of asset S3 under the four indicators are 2, 5, 5, 5, and after processing are 0.9899, 1, 1, 1, 1, and the corresponding weights of each indicator are (0.167, 0.227, 0.217, 0.389), and the importance α_3 of the business on asset S3 is obtained from formula (7) as:

$$\alpha_3 = w_1 * 0.9899 + w_2 * 1 + w_3 * 1 + w_4 * 1 = 0.9983$$

Similarly, we can get the importance of the business carried by other assets, see Table 6.

3.3 Value Assessment

Based on the method described in Sect. 2, the three major security attributes of the bank’s extranet assets are assigned, see Table 6. The confidentiality of assets is mainly analyzed and evaluated by the degree of disclosure of assets, for example, the confidentiality of deposit-related data within a bank is the highest, and once disclosed, it will have a very serious impact on the normal operation of the bank. The integrity is analyzed from the damage to the entire organization if the integrity of that asset is breached. The availability of assets is measured in terms of the damage caused to the organization by their functional interruptions. For assets which carry transaction-type services, there must be a connection channel with the bank’s internal network, so the confidentiality, integrity and reliability of the assets are of the highest level. For assets which run customer exchange services, there are generally connection channels established with the bank’s internal network. For assets that carry online investment services, there is a certain connection to the bank’s internal network. For assets carrying information service classes and other service classes, there is no connection channel with the bank’s internal network, so the C, I and A of the assets take lower values than the previous ones. For assets that carry other services, the CIA takes the lowest value compared to the others.

Table 6. Asset value indicators.

Asset number	C	I	A	Business importance	Asset number	C	I	A	Business importance
S1	5	5	5	1	S10	5	5	5	0.5206
S2	3	3	3	0.7083	S11	5	5	5	0.4970
S3	5	5	5	0.9983	S12	5	5	5	0.6676
S4	4	5	4	0.8038	S13	5	5	5	0.4683
S5	5	5	5	0.9966	S14	5	5	5	0.5830
S6	3	3	4	0.7049	S15	5	5	5	0.4396
S7	5	5	5	0.9949	S16	2	2	2	0.1502
S8	5	4	5	0.8977	S17	5	5	5	0.3974
S9	5	5	5	0.7115	S18	5	5	5	0.3890

From Table 6, the values of C, I, and A of asset S3 and the importance of the business it carries are 5, 5, 5, and 0.9983, respectively. Replacing formula (8), the value d_3 of asset S3 is obtained as:

$$d_3 = \sqrt[3]{0.9983 * 5 * 5 * 5} = 4.997$$

The remaining assets are evaluated by the same method and the results are written in Table 7.

Table 7. Comparison of asset value results.

Asset number	Traditional method	Methodology of this article	Asset number	Traditional method	Methodology of this article
S1	5	5	S10	5	4.022
S2	3	2.674	S11	5	3.961
S3	5	4.997	S12	5	4.370
S4	4	4.006	S13	5	3.883
S5	5	4.994	S14	5	4.177
S6	3	2.670	S15	5	3.802
S7	5	4.991	S16	2	1.063
S8	4	4.423	S17	5	3.676
S9	5	4.464	S18	5	3.650

3.4 Results Analysis and Comparison

For bank extranet assets, the comparison between the evaluation results obtained by this paper’s method and the traditional method which only considers the three major security elements is shown in Table 7. The ranking of asset values obtained by the traditional method [17] is from highest to lowest (S₁, S₃, S₅, S₇, S₉, S₁₀, S₁₁, S₁₂, S₁₃, S₁₄, S₁₅, S₁₇, S₁₈, S₈, S₄, S₂, S₆, S₁₆), where assets S₁ to S₁₅ and assets S₁₇, S₁₈ are all obtained with asset values of 5, and assets S₈, S₂, S₄, S₆ are all obtained with asset values of 3. The asset values obtained from the methods in this paper are ranked from highest to lowest (S₁, S₃, S₅, S₇, S₉, S₈, S₁₂, S₁₄, S₁₀, S₄, S₁₁, S₁₃, S₁₅, S₁₇, S₁₈, S₂, S₆, S₁₆), and the values of each asset are different. It is easy to conclude that the value of each asset calculated by the traditional method is the same, making it impossible to distinguish the value of assets that carry different business types in the same organization and assets that carry the same business type in different organizations, while the value of these assets can be clearly distinguished by the results calculated by considering the business importance factor proposed in this paper. For assets S₃ and S₄, which belong to the same organization but carry different types of business, the values of assets which are calculated by using the traditional method are 5 and 4, and the results obtained by using the method in this paper are 4.997 and 4.006. Both methods obtain a higher value for asset S₃ than asset S₄, which indicates that the proposed method is correct and feasible. For asset S₉ and asset S₁₂, which are both personal business systems, the values of C, I, and A are the same, so the results obtained by the traditional method are the same, both are 5, and it is impossible to distinguish whose value is higher, while the results obtained by using the method of this paper are 4.464 and 4.370, because although the CIA values of the two assets are the same, it can be seen from Table 7 that the importance of the business carried on S₉ is higher than that of S₁₂. This is because although the CIA values of the two assets are the same, the importance of the business carried on S₉ is higher than that of S₁₂. Therefore, it can be concluded that the value of asset S₉ is higher than that of asset S₁₂, which indicates that for assets which carry the same type of business

in different organizations, the value of these assets can be distinguished better by using the results calculated by this method than the traditional method.

It can be seen from the above examples that on the basis of the factors of confidentiality, integrity and reliability that affect the value of assets, it is necessary to consider the importance of the business carried by the assets, not only can reduce the influence of subjective factors, but also can solve the problem that the value of different assets carrying the same type of business cannot be distinguished by using traditional methods. Therefore, it is practical and realistic to use this paper's method to assess the value of assets.

4 Conclusions

The objective, accuracy, and ease of differentiation are the goals that must be achieved for information asset value assessment. In this paper, considering the three security attributes of confidentiality, integrity, and availability of assets, we propose that the value of assets is also influenced by the importance of the business they carry, and use the multiplication method to calculate the value of assets. We use the CRITIC assignment method to assign weights to four objective evaluation indicators which measure business importance: organization rank, organization level, service scope and business category, and then calculate business importance from the obtained weights of each indicator and the data processed by forward or inverse direction. In this paper, the feasibility of the proposed method is verified by evaluating the value of bank assets which are obtained from the extranet. The method can also be applied to other organizations to calculate the value of assets and prepare for the subsequent risk assessment work.

References

1. National Information Security Standardization Technical Committee.GB/T20984–2007 Information security technology information security risk assessment specification. Beijing: China Standard Publishing House (2007)
2. Tang, Z.C., Shen, Y-M.: An objective empowerment-based approach for assessing the importance of cyber assets. *Netw. Secur. Technol. Appl.* (04), 40–42 (2021)
3. National Information Security Standardization Technical Committee.GB/T31509–2015 Information security technology information security risk assessment specification. Beijing: China Standard Publishing House (2015)
4. Hong, X., Peng, F.: *Information Security Measurement and Risk Assessment*, pp. 222–225. Electronic Industry Press, Beijing (2012)
5. Zhou, J.X., Wang, S.-Q., Han, Y.-Y., Gu, Z.-J.: An information system security assessment model based on asset correlation. *Comput. Eng. Des.* **38**(07), 1691–1696+1718 (2017)
6. Zhou, Q.: *Research and Implementation of Business Process-Oriented Information Security Risk Assessment*. University of Defense Science and Technology (2008)
7. Wang, J.E., Kuang, X.-N., Qiu, J.J., Yang, J.-H., Zhang, F.: Crane safety evaluation based on comprehensive empowerment and improved set-pair analysis. *Crane Transp. Mach.* (23), 19–26 (2021)
8. Pan, B., Liu, S., Xie, Z., Shao, Y., Li, X., Ge, R.: Evaluating operational features of three unconventional intersections under heavy traffic based on critic method. *Sustainability* **13**(8), 4098–4127 (2021)

9. Krishnan, A.R., Kasim, M.M., Hamid, R., Ghazali, M.F.: A modified critic method to estimate the objective weights of decision criteria. *Symmetry*. **13**(6), 973 (2021)
10. Zhang, R.: Research on the Importance Assessment Method of Power Communication Network Nodes with Multi-Factor Evaluation Index. North China University of Electric Power (Beijing) (2020)
11. Zhang, Y.-T., Ji, X.-F.: Exploration of online banking security risk management. *Cooperat. Econ. Technol.* (09), 47–49 (2021)
12. Pisan.: bank classification ranking. *Internet Weekly* **2020**(08), 64–67 (2019)
13. Wang, S.: Research on the Development Strategy of G Bank's Online Banking Business. Heilongjiang University (2015)
14. Fan, J.: Exploration on the development of online banking business of commercial banks in China. *North. Finance* **09**, 43–46 (2020)
15. Zhang, Y.-P.: Analysis of Problems and Strategies of Online Banking of Commercial Banks in China. Tianjin University of Science and Technology (2020)
16. Liu, X.: Research on Countermeasures for the Development of Commercial Banks' Online Banking Business. Zhejiang University of Technology (2015)
17. Yang, L.-P., Xu, Y.-L.: The impact of two self-constructing functions on information security asset value assessment. *Comput. Modern.* (10), 102–105 (2013)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Vulnerabilities



Knowledge Graph Construction Research From Multi-source Vulnerability Intelligence

Lin Du^{1,2}(✉) and Chuanqi Xu¹

¹ Technical Team/Coordination Center of China, Tianjin Branch of National Computer Network Emergency Response, Tianjin, China

dulin@cert.org.cn

² School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China

Abstract. There is a huge Internet user group in China, and many enterprises and institutions are deeply affected by the threat of Cybersecurity vulnerabilities. At present, according to the needs of different business scenarios, relevant business personnel often need to search for different vulnerability information separately, relying on manpower, and the vulnerability intelligence distributed on the Internet has the characteristics of multi-source heterogeneity, which is difficult to ensure the effectiveness and reliability of vulnerability knowledge. In view of the above background, with vulnerabilities as the core, knowledge extraction of vulnerability intelligence is carried out according to existing standards, corresponding entities and relationships are established, and related and visualized knowledge graphs are studied and constructed to provide support for the discovery and traceability of vulnerability threats by information workers.

Keywords: Knowledge graph · Cybersecurity · Vulnerability · Named entity recognition · Relationship extraction

1 Introduction

With the rapid development of the Internet industry, a large number of Cybersecurity vulnerabilities have been gradually discovered and exploited in the use of various companies' products, causing potential risks to production and daily life. Vulnerability threat discovery and traceability have become common challenges and work requirements for personnel including system operation and maintenance and network management. There are various sources of vulnerability information, including vulnerability reports from various open source communities, public vulnerability databases, and products' patch information etc., which have the characteristics of scattered data, incomplete information, and different structures, and the vulnerability knowledge caused by data sources such as different Internet community platforms. The information of high quality and low quality are mixed, the repetition is high, the correlation is not clear, the data quality cannot be guaranteed, and it cannot effectively support the work needs of Cybersecurity business personnel for vulnerability detection, analysis and judgment.

In recent years, knowledge graphs can use deep learning to form valuable information and knowledge models through data collection, analysis, and mining. Since the knowledge graph theory was proposed by Google and applied to intelligent search [1], it was initially applied efficiently in the commercial field, such as the LinkedIn economic graph (User Profile) in the social field, and the Tianyancha enterprise graph (Enterprise Profile) in the field of enterprise information, etc.

In various vertical fields in China, there has been research and exploration on the application of knowledge graphs. An Ning et al. [2] proposed the construction of a cross-platform network public opinion knowledge graph, using Sina Weibo and Douyin short videos as data sources to build a network public opinion knowledge map, which is mainly used in the management and guidance of network public opinion. Xiao Le et al. [3] proposed knowledge graph for grain situation is mainly based on the grain situation dictionary and Flat-lattice model to extract grain situation entities for construction, which is used to assist grain situation decision-making. Mou Tianhao et al. [4] proposed a knowledge graph of process industrial control systems based on the control system cyber-physical asset management tasks to solve business problems related to industrial control systems. Zhang Kunli et al. [5] took obstetric diseases as the core and proposed a Chinese obstetric knowledge graph to facilitate medical question and answer and auxiliary diagnosis and treatment.

There are few applications of knowledge graphs in the field of Cybersecurity. This paper uses knowledge graphs to correlate numerous isolated vulnerability intelligences and present a panorama of vulnerability entities, which provides a new idea for vulnerability research and analysis, and helps to promote solutions for difficulties related to Cybersecurity business.

2 Vulnerability Knowledge Graph Construction Route

Large-scale domestic vulnerability databases include the China National Vulnerability Database (CNVD), the China National Vulnerability Database of Information Security (CNNVD) etc., which are the main methods for the construction and sharing of vulnerability intelligence [6]. Combining with the current situation of information security development, the sources of vulnerability intelligence in this paper are CNVD, CNNVD and CVE (Common Vulnerability Disclosure). After the vulnerability knowledge is integrated, manual proofreading is finally performed, and data with low confidence is discarded to ensure the quality of the vulnerability knowledge base. At the same time, the knowledge extraction model is continuously supervised and trained with new intelligence. With the accumulation of data, more new knowledge base data sources such as open source security websites are added as appropriate, and finally the entire system is iteratively updated.

2.1 Schema Layer Design

The schema layer of the vulnerability knowledge graph is above the data layer, and the core is the ontology library, which is an abstract representation of vulnerability knowledge, like the “class” in object-oriented. The schema layer mainly includes: entity-relation-entity, entity-attribute-attribute’s value. Based on “Information security technology—Cybersecurity vulnerability identification and description specification “ [8] (GB/T 28458–2020), the framework of vulnerability identification and description can be composed of identification items and description items. Taking into account the actual situation of domestic vulnerabilities, mainly from the perspective of vulnerability management and emergency response [9], the main attribute of the vulnerability is

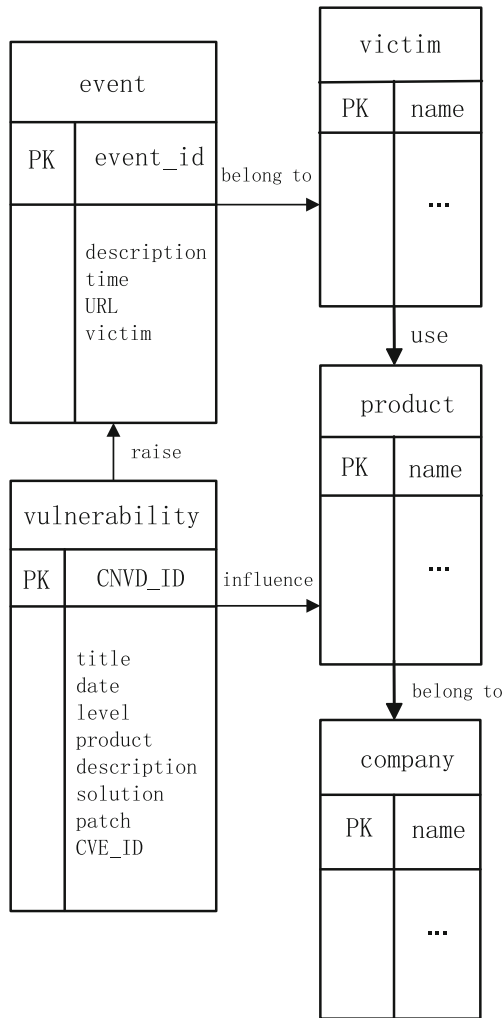


Fig. 1. The framework of entity and relationship

CNVD_ID, and the framework of the preliminary design entity and relationship is shown in Fig. 1.

Based on the graph structure, entities are used to represent objects or abstract concepts in the vulnerability space, and relationships are used to model inter-entity interactions, the framework follows the triplet of (head entity, relation, tail entity). Entities are distinguished by boxes, each row under the entity name has its attributes, PK represents the main attribute, and the arrow represents the relationship. The entity defines 5: vulnerability = {CNVD_ID, title, date, level, product, description, solution, patch, CVE_ID}; event = {event_id, description, time, URL, victim}; company = {name}; product = {name}; victim = {name}. Relationships define 4: influence, raise, belong to, use. More entities, attributes, and relationships can be gradually expanded according to this framework.

2.2 Data Layer Construction

The vulnerability knowledge graph data layer consists of three steps: data collection, knowledge extraction, and knowledge fusion.

2.2.1 Data Collection

Vulnerability, company, and product data are obtained from the unstructured text of the China National Vulnerability Database (CNVD) and semi-structured text of CVE (Common Vulnerability Disclosure) [10]. According to their own circumstances, the two entities, events and victims, can collect them in a compliant manner if they conduct unified management of vulnerabilities for the unit and its subordinate units, or as vulnerability managers.

2.2.2 Knowledge Extraction

Knowledge extraction is a method to automatically obtain structured information such as entities, relationships, and entity attributes from heterogeneous data such as semi-structured or unstructured data. According to the characteristics of vulnerability intelligence text, this paper marks the vulnerability intelligence text with BIOES [11], and then performs the following main operations: entity extraction, attribute extraction, and relation extraction. They are introduced as follows:

- 1) Entity extraction, namely named entity recognition (NER), refers to the automatic recognition of named entities from text datasets. At present, the main technical methods of named entity recognition are divided into: rule-based and dictionary-based methods -- manual construction of rule templates, and pattern and string matching as the main means; statistical-based methods -- including Hidden Markov Model (HMM), Maximum Entropy (MEM), Support Vector Machine (SVM), Conditional Random Field (CRF); Neural Network methods -- the main models are NN/CNN-CRF, RNN-CRF, LSTM-CRF. The goal of attribute extraction is to collect attribute information of a specific entity from different information sources. For example, for a specific vulnerability, attributes such as name and affected product can be obtained

from the public information on the Internet. Entity and attribute extraction this paper adopts the BLSTM-CRF model (Bidirectional Long Short-Term Memory Network - Conditional Random Field) [12], which is currently more effective in the field of security vulnerabilities, taking the product entity (Apache Log4j) as an example, as shown in Fig. 2

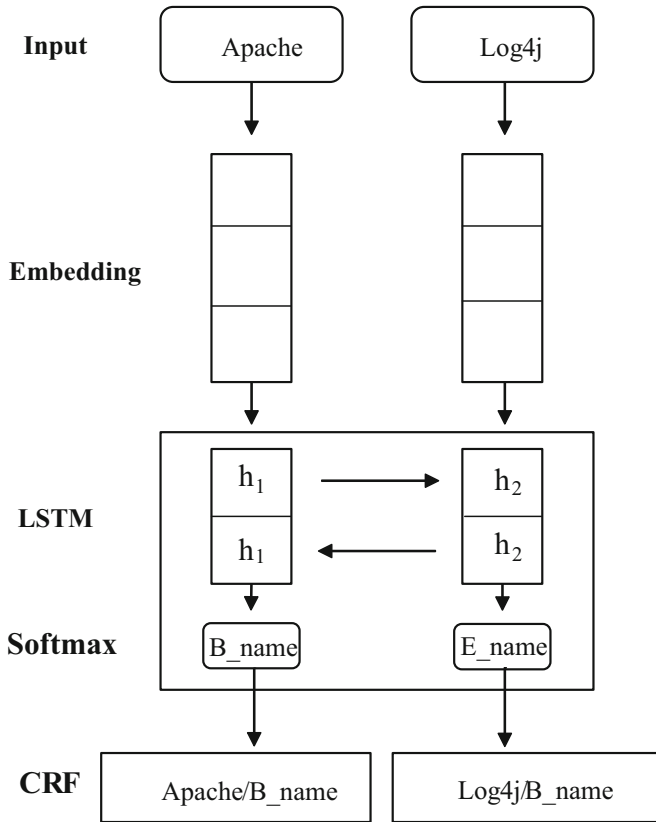


Fig. 2. The structure of BLSTM-CRF model

- 2) Relation extraction. After the vulnerability intelligence text is extracted by entities and attributes, a series of discrete named entities are obtained. Continuing to obtain semantic information requires relation extraction: extracting the interrelationships between entities from related texts, and connecting entities through relationships to form a networked knowledge structure. The vulnerability knowledge graph is different from the social character graph. The relationship is relatively small and simple. For example, vulnerability A “raises” event B. Since the relationship defined in the schema layer is easier to distinguish in text data such as vulnerability reports,

this paper chooses the method of rule matching, and the recognized entities are automatically selected according to the definition of the relationship in the category and schema layer, and fine-tuning is performed later. According to the definition, the entity can conform to the rules based on the pattern, so the relationship between the entities is determined according to the trigger word, and the designed rule samples are shown in Table 1.

Table 1. Samples of trigger word rules

No	Rules	Relation
1	Vulnerability A influences product B	Influence
2	Product B belongs to company C	Belong to
3	Vulnerability D raises event E	Raise
4	Event E belongs to victim F	Belong to
5	Victim F uses product G	Use

2.2.3 Knowledge Fusion

After data collection and knowledge extraction, entities, relationships and entity attribute information are obtained from the original unstructured and semi-structured vulnerability intelligence data. However, the relationship between multiple sources (information) is flat and lacks hierarchy and logic; there is still a lot of redundancy and misinformation in the knowledge. Knowledge fusion is to solve this problem, through entity disambiguation and coreference resolution, to realize the integration of vulnerability knowledge. For example, the company “苹果” and the “Apple” belong to the entity synonymous relationship and need to be integrated. After knowledge fusion, the noise and redundancy in the data are removed, and the quality of vulnerability knowledge is improved.

3 Vulnerability Knowledge Graph Construction Results

3.1 Experimental Environment

The experimental environment of this paper: the operating system is Windows 10; the CPU is AMD Ryzen™ 7 5800H@3.2 GHz; the GPU is GTX 3050Ti (4 GB); the memory is 64 GB; the Python version is 3.7; the neo4j version is 3.1.1.

3.2 Knowledge Graph Display

Taking some generic vulnerability data and a small number of influenced victims under Apache as an example (entities are vulnerability ontology, historical events, involved victims, companies, and products; relationships are the edges of a directed graph), the constructed visual interface is shown in Fig. 3.

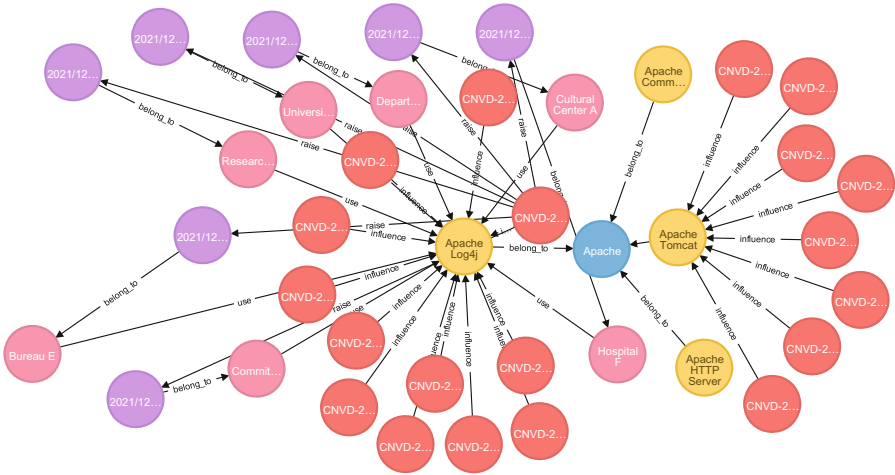


Fig. 3. Vulnerability knowledge graph

3.3 Application Analysis

In terms of vulnerability threat discovery and analysis, by constructing the graph to correlate and analyze vulnerability information, hidden information can be mined and effective judgments can be made. Referring to Fig. 3, various types of entities are used as nodes in the graph, and various types of relationships between entities are used as edges in the graph. Starting from a certain entity, such as a victim with critical infrastructure, you can know which products of which companies are used by the victim, and which security events have occurred due to which vulnerabilities occurred at specific times. Once a 0-day vulnerability occurs again in the corresponding products of the company, it can be reasonably predicted that the victim will be influenced by this vulnerability, and it will be warned in time before possible Cybersecurity events to avoid major losses. This information is often unavailable from a single vulnerability report, and knowledge graphs can organically connect numerous vulnerability information.

4 Conclusion

According to the characteristics of the vulnerability field, this paper first integrates multi-source vulnerability intelligence data to design a vulnerability knowledge graph framework; then uses a deep learning model to extract entities and attributes, extracts relationships based on pattern rules, and constructs a vulnerability knowledge ontology, check and analyze; and finally complete the multi-source knowledge graph. In the future, by further adding multiple vulnerability threat intelligence data sources, a larger and more complete vulnerability knowledge graph can be formed, which can effectively provide more Cybersecurity decision support for information workers.

References

1. Singhal, A: Introducing the knowledge graph: things, not strings. *Offic. Google Blog* **5**, 16 (2012)
2. An, N., An, L.: Construction and comparison of cross-platform online public opinion knowledge graph. *Inf. Sci.* **40**(03):159–165 (2022). (in Chinese)
3. Xiao, L., Li, J. X., Ge, L., et al.: Knowledge graph construction for decision support of grain situation. *J. Chin. Cereals Oils Assoc.* **03**(05), 1–14 (2022). (in Chinese)
4. Mou, T.H., Li, S. Y.: Knowledge graph construction for control systems in process industry. *Chin. J. Intell. Sci. Technol.* **4**(01), 129–141 (2022). (in Chinese)
5. Zhang, K.L., Hu, C.X., Song, Y., et al.: Construction of Chinese obstetric knowledge graph based on multiple source data. *J. Zhengzhou Univ. Natl. Sci. Ed.* **05**(30), 1–7 (2022). (in Chinese)
6. Dong, C., Jiang, B., Lu, Z.G., et al.: Knowledge graph for cyberspace security intelligence: a survey. *J. Cyber Secur.* **5**(05), 56–76 (2020). (in Chinese)
7. Ji, S.X., Pan, S.R., Cambria, E., et al.: A Survey on Knowledge Graphs: representation, acquisition and applications. *arXiv preprint arXiv:2002.00388* (2021)
8. Committee, N.I.S.S.T.: Information Security Technology—Cybersecurity Vulnerability Identification and Description Specification: GB/T 28458–2020. Standards Press of China, Beijing (2020).(in Chinese)
9. Committee, N.I.S.S.T.: Information Security Technology—Specification for Cybersecurity Vulnerability Management: GB/T 30276–2020. Standards Press of China, Beijing (2020).(in Chinese)
10. Zhou, W.P., Yang, W.Y., Wang, X.H., et al.: Research on penetration testing tool for industrial control system. *Comput. Eng.* **45**(08), 92–101 (2019). (in Chinese)
11. Reimers, N., Gurevych, I.: Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *arXiv preprint arXiv:1707.06799* (2017)
12. Zhang, R.B., Liu, J.Y., He, X.: Named entity recognition for vulnerabilities based on BLSTM-CRF model. *J. Sichuan Univ. Natl. Sci.* **56**(03), 469–475 (2019). (in Chinese)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Mobile Internet



Research and Practice of TCP Protocol Optimization in Mobile Internet

Shizhan Lan^{1,2(✉)}, Lifang Ma³, and Jing Huang⁴

¹ School of Software Engineering, South China University of Technology, Guangzhou 510006, China

lanshizhan@gx.chinamobile.com

² China Mobile Guangxi Branch Co., Ltd., Nanning 530012, China

³ Guangxi Polytechnic of Construction, Nanning 530007, China

⁴ Cloud and Network Strategy Design Department, EVERSEC (Bei Jing) Technology Co., Ltd., Beijing 100191, China

Abstract. This paper analyzes the problems of traditional TCP protocol in the wireless network environment and proposes a scheme based on performance-enhancing agents, which is more suitable for the actual situation of current wireless core networks. TCP application optimization was employed to enhance congestion control. Based on the automatic learning mechanism of network path features, this paper proposes herein the dynamic algorithm ZetaTCP. In practice, the performance enhancement agent based on ZetaTCP. In practice, the performance enhancement proxy based on ZetaTCP was verified and achieves good results in LTE networks. In practice, the performance enhancement proxy based on ZetaTCP was verified and achieved good results in the LTE network.

Keywords: LTE · TCP protocol optimization · Congestion control · ZetaTCP

1 Introduction

TCP protocol, As the main protocol of online data transmission, TCP protocol has been widely used on mobile Internet, which carries over 90% of mobile Internet traffic in this case. Though the LTE network advances and the mobile Internet notably speeds up, traditional TCP technology tailored for the wired network environment cannot adapt to the wireless network environment with relatively poor link quality and frequent changes in latency and packet loss. Therefore, the improvement of the transmission performance of TCP protocol in the wireless network and the enhancement of the bandwidth utilization of wireless links are critical to optimizing the wireless core network [1].

2 Features and Problems Analysis of Standard TCP

TCP provides transmission services with reliable point-to-point connections, subject to sliding windows to control the transmission rate. The congestion control of standard

TCP contains several key technologies, including “slow start”, “congestion avoidance”, “fast retransmission”, “quick recovery”, and “retransmission timeout” [2].

Standard TCP is very sensitive to packet loss, halving or minimizing the value of the congestion windows with a slow increase. Previously, packet loss in the wired network often indicates the occurrence of network congestion, which can be better controlled and quickly recovered by standard TCP. Nonetheless, the current wireless network environment brings about increasingly salient defects of the standard TCP protocol:

2.1 Ineffective Congestion Judgment Mechanisms

In a wired network with a low bit error rate, it is rational for TCP to assume that packet loss is triggered by network congestion. The packet loss, nevertheless can also be caused by sudden errors in wireless channels, mobile device handoffs, attenuation channels, or changes in network topology. In this context, standard TCP cannot accurately distinguish whether packet loss is derived from congestion or not, resulting in congestion misjudge.

2.2 Slow Congestion Recovery Mechanisms

Once detecting packet loss, TCP will trigger the response for congestion control in three steps. At first, the packets failing to be confirmed will be retransmitted, thus reducing the congestion window and the transmission rate; Then, it will activate the congestion control mechanism, consisting of exponential back-off of the timeout clock and a reduction in the slow start threshold. At last, the congestion avoidance stage will be activated to relieve the congestion. If the packet loss results from channel errors or mobile device handoffs, the congestion recovery mechanism of TCP will induce throughput drop and longer latency.

2.3 Inaccurate Packet Loss Judgment Mechanism

The standard TCP stack determines packet loss by two methods. One is the number of consecutive Dup-ACKs, and the other is the ACK timeout. When there are considerable packet losses, ACK timeout is preferred to interpret the timeout and trigger retransmission. In a modern network, packet losses are often burst, and it's natural that multiple data packets are lost simultaneously on a connection. Therefore, standard TCP must rely on timeout for retransmission, which often leads to a waiting state of several or even ten seconds, causing long stagnant transmission, or even disconnection [3].

3 ZetaTCP Optimization

Pursuant to the survey and analysis of the quality of mobile Internet access in all network operators, TCP traffic accounts for the vast majority of all the existing network traffic of mobile users accessing mobile Internet applications. However, due to the frequent and changing delay and packet loss of the wireless network environment, the transmission efficiency of traditional standard TCP is often substantially low in this context [4]. In case the defects of TCP's treatment mechanism in various wireless network conditions can be corrected, and the efficiency of TCP traffic transmission can be enhanced, the user's mobile Internet experience can be significantly improved [5].

3.1 Comparison and Improvement of TCP Optimization Technology

Most domestic and overseas TCP protocol optimization technologies apply static algorithms, which utilize fixed congestion judgment and recovery mechanisms in accordance with the assumption of the Internet traffic model. As the Internet environment progresses, the traffic characteristics are increasingly complicated and difficult to predict. Against such a circumstance, these TCP protocol optimization techniques can only be valid in specific network scenarios where the premise is established. Moreover, as the transmission progresses, the network path characteristics may change and the effect may turn out to be unstable. Two common TCP optimization algorithms are presented below:

The Vegas TCP algorithm defines a state variable, Base RTT (basic round-trip delay), whose theoretical value should be “round-trip delay of connection without congestion.” With the delay change as the congestion indicator, the Vegas algorithm is more sensitive to the judgment of network congestion so as to decrease the packet loss rate of the network and obtain excellent average throughput rates in all networks using the Vegas algorithm. Nevertheless, in a network environment mixed with packet loss-based algorithms, it has always seen a rapid rise in time delay occurring before packet loss. In this case, Vegas always shrinks CWND (congestion window) before packet loss-based algorithms and reduces the transmission rate, making its overall performance inferior to packet loss-based algorithms. Vegas TCP is characterized by the relatively low transmission performance during shallow cohort congestion and frequent changes in wireless network delay [6].

CUBIC TCP, an enhanced version of BICTCP, simplifies its window adjustment algorithm. A cubic function is deployed as the growth function of the congestion window, grows only according to on the time interval between two consecutive congestion events. CUBIC is the default TCP algorithm of the Linux kernel. The CUBIC TCP has relatively low transmission performance under non-congestion packet loss and deep queue congestion in wireless networks [6].

The Performance-enhancement Proxy Based Scheme is adopted in a bid to improve the drawbacks of the above algorithms and enable TCP to register a high transmission efficiency in the wireless network with long latency and frequent link errors [4]. The method to segment the original TCP connection by the above Scheme is also known as TCP segmentation (see Fig. 1).

The Performance-enhancement Proxy Based Scheme follows the idea that local problems should be solved locally. By deploying the agent, the TCP connection between the server and the wireless mobile end falls into two sections at a certain node in the middle, with one deployed on the server sending end of the fixed network, and the other is connected to the mobile receiving end of the wireless network, which blocks the influence of the wireless environment on the server sending end. In this case, the server sending end can prevent irrational activation of the congestion control algorithm irrespective of random packet loss of the wireless network. By virtue of the improved TCP deployed on the enhanced proxy, the performance of TCP in wireless networks will be strengthened, and data transmission rate to mobile ends will be elevated [7]. In this scheme, there is no need for any modification on the TCP protocol stack of the server sending end and the wireless mobile receiving end, which is feasible to implement at this stage.

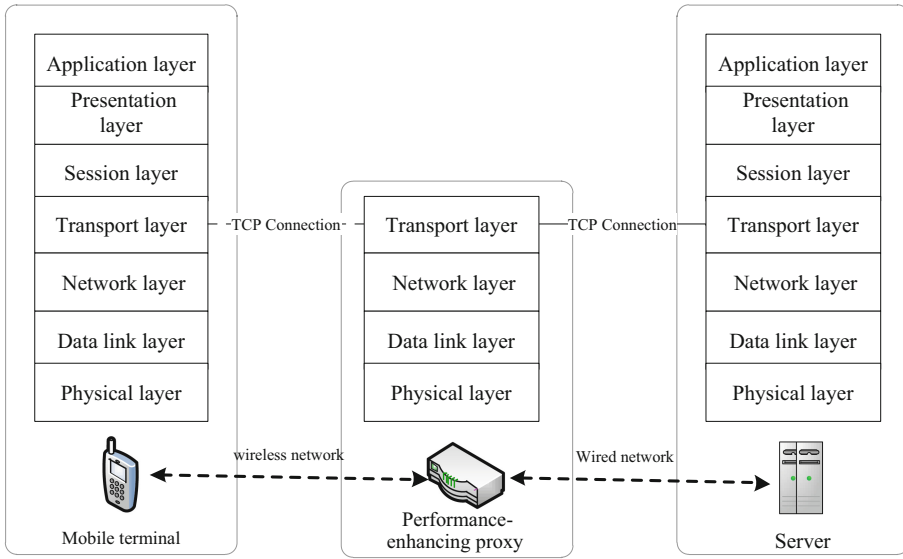


Fig. 1. Schematic diagram of performance-enhancement proxy based scheme

3.2 ZetaTCP Optimality Principle

It adopts the dynamic self-learning algorithm (ZetaTCP) with network path features and utilizes the Performance-enhancement Proxy Based Scheme. It observes and analyzes the real-time network features on each TCP connection and adjusts the algorithm anytime in accordance with the learned network characteristics. By doing so, it can judge the degree of congestion more accurately and distinguish packet loss more promptly, thereby handling the congestion more appropriately and retransmitting the lost packet more swiftly. According to the design principle, it helps the static algorithms adapt to changes in network path characteristics and ensures that the acceleration effect is constantly valid even under various network environments and frequently changing network delay and packet losses.

Besides applying the above two approaches of standard TCP, ZetaTCP also considers packet loss and delay changes into consideration and introduces a self-learning dynamic algorithm mechanism with TCP connection path network characteristics to make the congestion judgment more accurate and timely. The dynamic learning mechanism can be used to determine the network path characteristics of each specific connection during the transmission process. The characteristics include end-to-end delay and its changing features, arrival interval and its variation of receiving end feedback packet (ACK), packet reversal degree and its changing features, delay jitter possibly caused by deep data detection of security equipment, and random packet loss induced by various factors. ZetaTCP tracks these features in real time, apprehends these features in all aspects and deduces the precursor signals reflecting congestion and packet loss on this specific TCP connection network path. With the above steps, the congestion degree and congestion recovery mechanism appropriate for the available bandwidth of the current path, and

packet loss judgment and recovery can be determined in light of these dynamic intelligent learning results and the transmission rate.

ZetaTCP is implemented by an automatic learning state machine (Learning State-Machine), as indicated in Fig. 2.

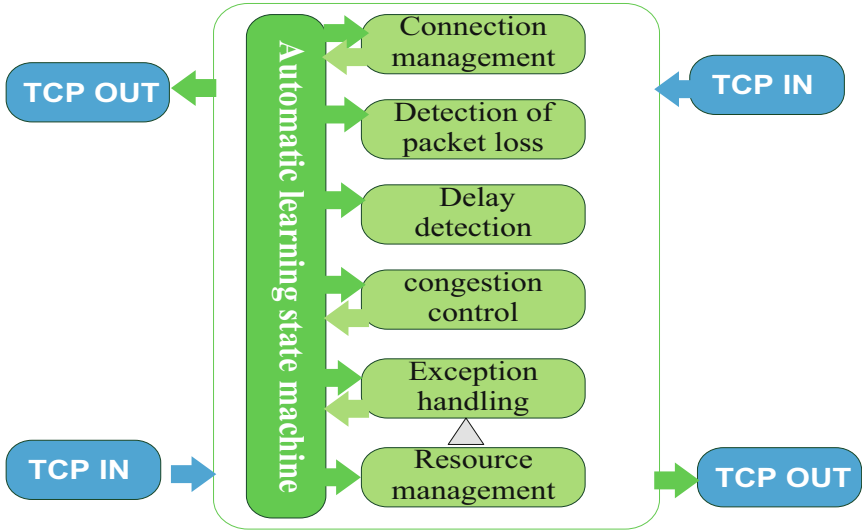


Fig. 2. ZetaTCP automatic learning state machine

Each automatic learning state machine matches a TCP connection, records the network path characteristics of the TCP connection, and dynamically determines appropriate congestion judgment, recovery mechanism and the packet loss judgment mechanism. In specific, connection management can directly extract the external features of the network path and input them to the machine. The intelligent learning outcomes accumulated by this machine are subject to the packet loss monitoring, congestion control, exception handling and delay monitoring modules to adjust the transmission behavior of the corresponding TCP connection. The dynamic feedback can be conveyed to the automatic learning state machine through the exception handling and congestion control modules to optimize network path learning further.

3.3 Implementation Algorithm of ZetaTCP Optimization

On Linux, Netfilter is enabled for packet interception. In different deployment scenarios, Netfilter can perform Hooks at the Ethernet bridge level to implement the transparent bridge mode or conduct Hooks at the INET level to perform the routing mode. A pair of Hook points can be mounted to the LAN and WAN of the engine on NF_INET_POST_ROUTING and NF_INET_PRE_ROUTING respectively as shown in Fig. 3.

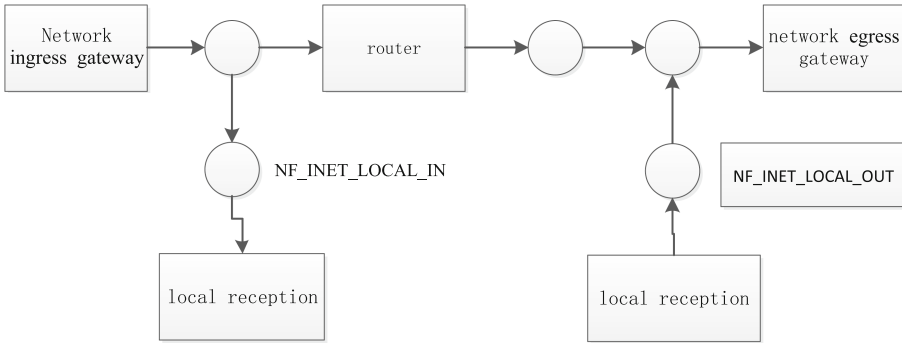


Fig. 3. Implementation of Hook point of ZetaTCP

The ZetaTCP’s congestion control algorithm is available, as shown in Fig. 4.

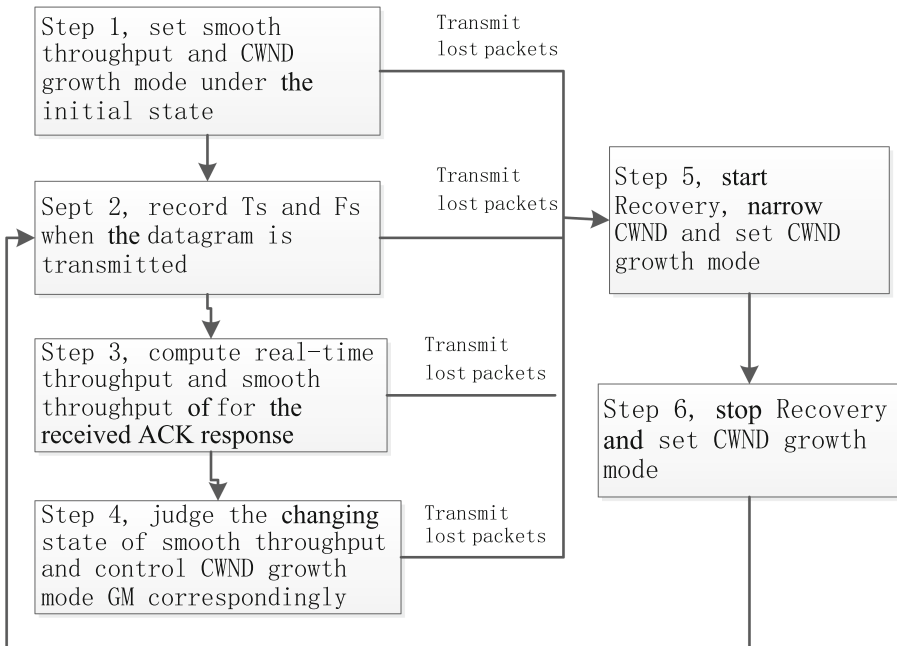


Fig. 4. Process flow of the congestion control algorithm of ZetaTCP

For the data message with the highest sequence number in the received ACK response, the actual instant throughput rate is calculated as per $BC = FS / (T - TS)$. Wherein, T denotes the current time, TS indicates the sending time of the data packet with the highest sequence number, and FS means as the total amount of data sent at this TS time and fails to be responded to by ACK. The said TS and FS are recorded when the data packet with the highest sequence number is sent.

The smooth throughput rate is determined according to $B = (1 - \alpha) * B' + \alpha * BC$; Wherein, α refers to a constant parameter, BC means the actual instant throughput, and B' suggests the last calculated smooth throughput rate.

CWND growth modes fall into categories of exponential growth, linear growth, and stop. Provided that the increase in smooth throughput rate exceeds the previous smooth throughput rate, the CWND growth mode is set as exponential growth. If the smooth throughput rate declines continuously for a predetermined number of times, and the total amount of smooth throughput rate drops is not less than the throughput rate drop threshold, it is required to judge further whether the current smooth round-trip delay $SRTT$ is less than or equal to $\eta * RTTMIN$; Wherein, $RTTMIN$ means the smallest round-trip delay, and η refers to a constant parameter; If yes, the CWND growth mode should set as linear growth; If not, it shall be set to stop.

ZetaTCP can, through the foregoing algorithm, obtain the real-time optimal CWND value, thereby maximizing network throughput and preventing congestion.

4 Application of ZetaTCP Optimization Technology in Mobile Internet

The packet losses occurred wireless networks are often attributed to signal loss, interference and other causes, and the packet loss rate therein is greater than that in wired networks [8]. In consequence, standard TCP usually fails to judge these packet losses in a quick manner, hence bringing about low transmission efficiency, unstable transmission quality, high unpredictability, and poor user experience. By contrast, ZetaTCP can quickly predict packet loss and recover in time in a wireless network environment, making the transmission more stable and quicker, thereby considerably improving the user experience.

In order to verify the actual effect of the ZetaTCP optimization technology in the wireless core network, the Performance-enhancement Proxy Based Scheme and the LotWan acceleration system using ZetaTCP as the proxy node are introduced to optimize the data transmission of the wireless core network and evaluate its optimization effect.

4.1 ZetaTCP Optimization Deployment Scheme

As indicated in Fig. 5, the ZetaTCP acceleration device is deployed outside the SGi port of the PDN-GW, which is transparently connected in series between the PDN-GW and the firewall or on the Internet side of the firewall. The acceleration device is transparently connected to the network and works as a TCP proxy to accelerate the coverage of the whole wireless network transmission path.

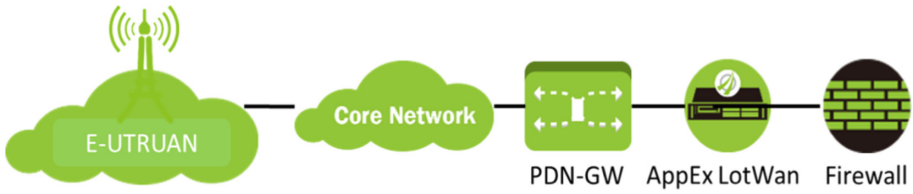


Fig. 5. ZetaTCP acceleration device deployment scheme

4.2 ZetaTCP Optimization Application Results

The implementation environment selected here covers three kinds of wireless networks with poor areas: medium-low field strength coverage, hotspot, and busy-hour regions. The data shows, the congestion control algorithm adopted by ZetaTCP in these three areas achieves relatively faster transmission speeds. The average results of application assessments are listed in the following table (Tables 1, 2 and 3).

Table 1. Acceleration effect of web browsing services

Web browsing services	Time delay after acceleration (s)	Time delay without acceleration (s)	Delay variation	Throughput after acceleration (KB/s)	Throughput without acceleration (KB/s)	Promotion of throughput
Sina	12.66	18.62	31.99%	425.55	261.95	62.45%
Sohu	5.57	11.77	52.67%	109.86	65.70	67.22%
NetEase	1.44	26.68	45.86%	117.84	67.40	74.83%
Mobile phone Tencent	7.93	14.10	43.71%	94.85	67.81	39.89%
Mobile games	6.37	13.41	52.45%	123.99	82.96	49.46%
People.cn	18.46	33.90	45.54%	52.91	30.87	71.37%
STO express	5.47	9.73	43.75%	133.51	89.56	49.06%
Harvest fund	17.97	29.61	39.30%	112.24	64.48	74.07%
CCB	6.51	11.10	41.37%	62.89	40.45	55.47%
Xinhuanet	5.39	10.12	46.68%	154.48	95.84	61.18%
Overall (Average value)	—	—	44.33%	—	—	60.50%

Table 2. Acceleration effect of file download services

File download	Rate after acceleration (KB/s)	Rate without acceleration (KB/s)	Promotion of throughput
Client terminal of QQ - 23.5M	2851.21	1410.61	102.13%
Client terminal of MM shopping mall - 12.9M	2362.47	1159.66	103.72%
Overall (Average value)			102.92%

Table 3. Acceleration effect of video download services

Video download	Rate after acceleration (KB/s)	Rate without acceleration (KB/s)	Promotion of throughput
Sohu video	2365.09	1288.95	83.49%
Youku tudou	2186.83	1127.95	93.88%
Overall (Average value)			88.68%

On the basis of the above results, due to the delayed judgment of these packet losses, the transmission efficiency of standard TCP is often low with unstable transmission quality, which is difficult to predict and seriously affects the user experience. As for ZetaTCP acceleration in the wireless network environment, the corresponding connection network characteristics can be accumulated through dynamic learning. Making the ZetaTCP congestion control algorithm more accurate. It also can predict packet loss very quickly and recover in time, making the transmission more stable and quicker, thereby significantly improving the user's experience.

5 Conclusion

TCP optimization, an essential approach for telecom operators to optimize the wireless core network, remarkably boosts the Internet access rate of mobile phone users, enhances user perception, and adds to the competency of traffic management.

This paper proposes, an improvement scheme of ZetaTCP performance in the wireless network environment is proposed by combining it with the enhanced TCP congestion control mechanism and applying it in the production environment of the current network. The experimental results demonstrate that ZetaTCP is a good guarantee for TCP users to get the appropriate bandwidth as defined by the flow specification. It can eliminate the unfairness problem caused by different RTTs when congestion occurs. It both maintains the end-to-end semantics of TCP, and takes corresponding measures upon distinguishing the types of network packet loss, thereby bolstering the transmission performance of TCP.

References

1. Low, S.H., Paganini, F., Doyle, J.C.: Internet congestion control. *IEEE Control Syst. Mag.* **22**(1), 28–43 (2002)
2. Low, S.H., Peterson, L.L., Wang, L.: Understanding TCP Vegas: a duality model. *J. ACM (JACM)* **49**(2), 207–235 (2002)
3. Mo, J., Walrand, J.: Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw. (TON)* **8**(5), 556–567 (2000)
4. Srikant, R.: *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications)*. Springer, Heidelberg (2004)
5. Wen, Y., Shi, Z.: Research on congestion control mechanism of wireless network. *Comput. Eng. Sci.* **26**(10), 27–29 (2004)
6. Perkins, C.E.: IPM obility support. RFC2002 (October 1996)
7. Cheng, J., Wei, D.X., Low, S.H.: FAST TCP: motivation, architecture, algorithms, performance. In: *Proceedings of IEEE INFO-COM2004 Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies* (2004)
8. Floyd, S.: HighSpeed TCP for Large Congestion Windows. RFC 3649 (2003)
9. Bisu, A.A., Gallant, A., et al.: Experimental performance evaluation of TCP over an integrated satellite-terrestrial network environment. In: *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 781–786 (2019)
10. Cardwell, N., Jacobson, V., Cheng, Y., et al.: Model-based network congestion control. Technical Disclosure Commons (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Traffic Analysis



Research on Adversarial Patch Attack Defense Method for Traffic Sign Detection

Yanjing Zhang¹, Jianming Cui¹, and Ming Liu²(✉)

¹ School of Information Engineering, Chang'an University, Shaanxi 710064, China

² National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
liuming@cert.org.cn

Abstract. Accurate and stable traffic sign detection is a key technology to achieve L3 driving automation, and its performance has been significantly improved by the development of deep learning technology in recent years. However, the current traffic sign detection has inadequate difficulty resisting anti-attack ability and even does not have basic defense capability. To solve this critical issue, an adversarial patch attack defense model IYOLO-TS is proposed in this paper. The main innovation is to simulate the conditions of traffic signs being partially damaged, obscured or maliciously modified in real world by training the attack patches, and then add the attacked classes in the last layer of the YOLOv2 which are corresponding to the original detection categories, and finally the attack patch obtained from the training is used to complete the adversarial training of the detection model. The attack patch is obtained by first using RP_2 algorithm to attack the detection model and then training on the blank patch. In order to verify the defense effective of the proposed IYOLO-TS model, we constructed a patch dataset LISA-Mask containing 50 different mask generation patches of 33000 sheets, and then training dataset by combining LISA and LISA-Mask datasets. The experiment results show that the mAP of the proposed IYOLO-TS is up to 98.12%. Compared with YOLOv2, it improved the defense ability against patch attacks and has the real-time detection ability. It can be considered that the proposed method has strong practicality and achieves a tradeoff between design complexity and efficiency.

Keywords: Traffic sign detection · Adversarial patch attack · Deep learning

1 Introduction

Traffic sign detection is a key technology that is continuously updated and iterated in the vision-based advanced driver assistance systems. Its purpose is to establish accurate, real-time and safe traffic sign recognition capabilities for complex and dynamic real roads [1]. The most widely used technology is target detection based on Deep Neural Networks (DNN) [2]. However, many recent studies have shown that the security of DNN models is not reliable, that is, it is susceptible to the influence of adversarial examples, which would mislead the classifier produces incorrect predictive output [3–5]. Currently, adversarial patch attacks in the physical world have been considered as a very effective

means for attacking object detection models, and have achieved remarkable results in the fields of image classification [6], face recognition [7], object detection and etc. [8–10]. In order to deal with the security threats caused by patch attacks, a growing number of researchers began to study defense methods. However, current researches mainly focus on image classification, and there are few reports on traffic sign detection. In addition, traditional image pre-processing methods, such as image denoising [11], local gradient smoothing [12], and partial occlusion [13], would reduce the detection accuracy on the original samples, and most of them are designed to operate in the digital space and are ineffective to the physical world.

YOLO (You Only Look Once) series is a one-stage object detector that can directly output bounding boxes and categories. Compared with RCNN (Region-Convolutional Neural Networks), Faster-RCNN and other two-stage networks, YOLO has a lighter structure, fewer parameters, and faster speed. Therefore, it is more suitable for application research in the field of automatic driving that requires high real-time and accuracy [14]. Compared with v3–v5, YOLOv2 has less computation in forward reasoning [15–18], and can maintain a relatively high mAP (mean Average Precision) in the COCO dataset test under the same scale input. In addition, in automatic driving, object detection models are mostly deployed on edge devices for inference, resulting in limited model storage space and computing resource [19]. YOLOv2 mainly consists of convolutional layers and softmax, which is easier to implement in mobile device and can also accelerate inference by small graphics cards. Therefore, the interesting and challenging question addressed here is how to integrate and extend YOLOv2 to traffic sign detection and achieved the stable defense capability.

To solve the above problems, we propose an adversarial patch defense model IYOLO-TS (Improved YOLOv2 on Traffic Signs) on traffic sign detection. The main contributions can be summarized as follows: (1) We extend the research of patch attack defense to the field of traffic sign detection and proposed a practical defense model IYOLO-TS. (2) We improved the last layer of YOLOv2 model by adding an additional 11 attacked classes, and optimized its structure to ensure the high detection performance for normal traffic signs. (3) In order to achieve high robustness and more realistic style against perturbations, we adopt RP₂ algorithm [8] to attack the YOLOv2 and pioneered the development of a patch dataset named LISA-Mask.

2 Improved YOLOv2 on Traffic Signs Detection Model

2.1 Framework Design of IYOLO-TS

Figure 1 provides an overview of IYOLO-TS. From the structure of the neural network, IYOLO-TS adds 11 additional attacked categories to the last softmax layer. As a result, IYOLO-TS is able to detect the attacked targets while accurately identify the attacked targets to the true classes, which are defined as the right part of Fig. 1.

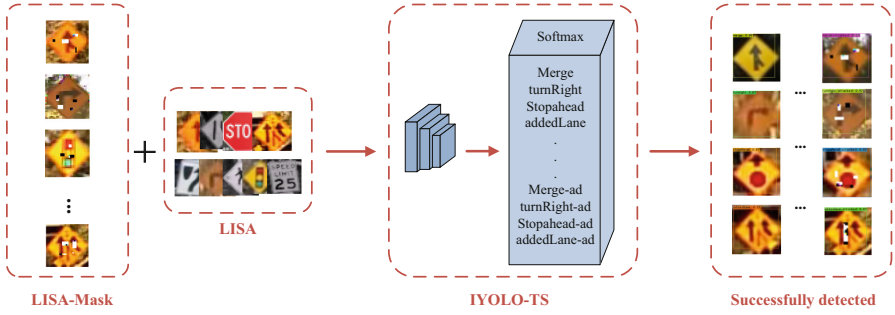


Fig. 1. Framework of IYOLO-TS.

We sample from each category of LISA and LISA-mask to train IYOLO-TS. IYOLO-TS retains the network structure of yolov2 except for the final softmax layer by adding 11 attacked categories. The right part of figure is attacked traffic sign detection result of IYOLO-TS. The base idea of YOLOv2 is to represent the output of the feature map as the center, width and height of the bounding box, as well as the confidence and category. YOLOv2 divides the input image x into N preselected areas, and each area predicts M anchor box. Assuming that there are n classes to be identified, for the LISA and LISA-Mask datasets, n is 11, each anchor box can be written as an $(n + 5)$ dimensional vector. The result of the feature map for each anchor box can be expressed as shown below:

$$\langle \hat{X}, \hat{Y}, W, H, P_{obj}, P_{cls1}, \dots, P_{clsn} \rangle \tag{1}$$

where \hat{X}, \hat{Y}, W, H are the center and size of bounding box, P_{obj} is the confidence score indicates the probability of whether the bounding box contains a target and P_{cls_i} is class score. Then, arrange the anchor boxes in order, and each preselected area would output a vector with dimension $M(n + 5)$. Eventually, the output of YOLOv2 is a vector of dimension $NM(n + 5)$. IYOLO-TS inherits the form of the YOLOv2 loss function and adds the loss to the attacked class score. We add 11 attacked categories to the last softmax layer of YOLOv2, so the length of each anchor boxes vector becomes $(n + 5 + 11)$, and the corresponding final output becomes a vector of $NM(n + 5 + 11)$ dimension. This gives IYOLO-TS two advantages: the detection speed inherited from YOLOv2 meets the time-sensitive requirements for defending against physical world attacks and can also be used as a model for detecting attacks.

2.2 RP₂-Based Attacking Process

In order to achieve a high robustness and a more realistic style against perturbations, we use the method in [8] to attack the YOLOv2 detectors. To generate visual adversarial perturbations that are robust under different physical conditions, RP₂ algorithm is first derived without considering other physical conditions, starting with the optimal method for generating perturbations to a single image x . Then update the algorithm considering continuous changes in the distance and angle of the camera to the road sign. Then, the

constrained optimization problem of RP_2 is expressed as below:

$$\arg \min_{\delta} \lambda \|\delta\|_p + J[f_{\theta}(x + \delta), y^*] \quad (2)$$

where $J(\cdot)$ is the loss function measures the degree of difference between the prediction of the model and the target class y^* . x is the input, δ denotes the perturbation of input x , $f_{\theta}(\cdot)$ denotes the target classifier, and λ is the hyperparameter that controls the regularization of the distortion. Specifying the distance function as $\|\delta\|_p$, which denotes the p -norm of δ . To better capture the effects of changing physical conditions, partial experimental samples containing random noise are generated to be added to the algorithm iterations. To ensure that the perturbation is applied only to the surface of the target object, a mask is introduced that will limit the physical region of the perturbation. The final robust spatially constrained perturbation is optimized as:

$$\arg \min_{\delta} \lambda \|M_x \cdot \delta\|_p + NPS + E_{x_i \sim X^v} J\{f_{\theta}[x_i + T(M_x \cdot \delta)], y^*\} \quad (3)$$

where the matrix M_x is the representation of the mask, NPS is the unprintability fraction, and the function $T(\cdot)$ represents the alignment function that maps the transformation of the object and the perturbation. Since all perturbation values must be reproducible in the physical world and there exist some reproduction errors in the colors produced by the printer [20], RP_2 adds an additional term NPS to the objective function to model the printer color reproduction errors. It can be found that during an attack, forged patches generated under the qualification of different masks can simulate common vandalism behaviors that are ignored by most people. Such attacks in the physical world are highly disruptive to traffic sign detectors, so it is imperative to develop appropriate defense strategies.

2.3 Generating of LISA-Mask Dataset

In order to make IYOLO-TS more generalizable and make it effective in defending against various patch attacks, we generate 50 different masks and constructs a new dataset named LISA-Mask to help train the IYOLO-TS.

During attack patches generating experiment, we found that the patches at different locations have an impact on the effectiveness of the attack, and each mask produces a different attack effect. In addition, in order to simulate a more realistic random attack scenario as much as possible, 50 different masks are produced in this paper by limiting the size, distance, number and shape of the scope. The generated masks are different from other target detection datasets that can take the whole area as the area of interest for the attack, the masks in this paper should limit the size of the scope so that they avoid obscuring the whole pattern of traffic signs.

The success rate of the attack can be expressed as follows:

$$\frac{\sum_{c \in C} \{f_{\theta}[A(c^{d,g})] = y^* \wedge f_{\theta}(c^{d,g}) = y\}}{\sum_{c \in C} [f_{\theta}(c^{d,g}) = y]} \quad (4)$$

where $A(c^*)$ represent a set of images with incorrect classification results from original images set c . $c^{d,g}$ represent the images taken from distance d and angle g . Respectively,

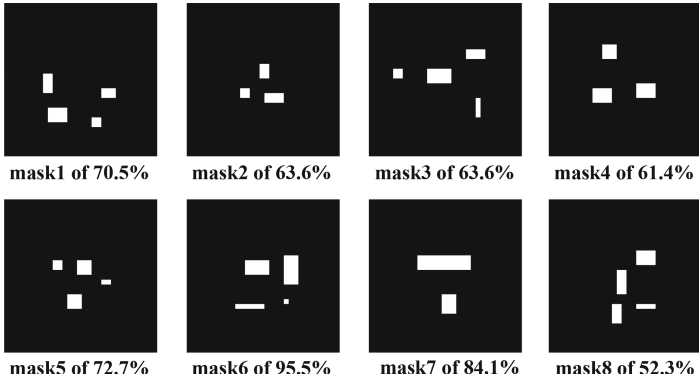


Fig. 2. Some of the masks and their attack success rates.

y is the actual class label of the target, and y^* is the detection result of the target after the attack. As shown in Fig. 2, some of the generated masks and their attack success rates. It can be seen that different kinds of masks can lead to different degrees of reduction in YOLO’s inference results, i.e., physical attacks on traffic signs can be simulated to some extent.

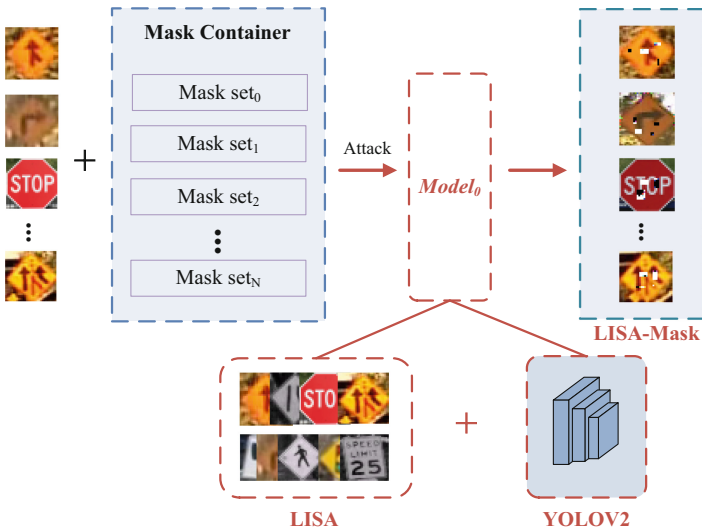


Fig. 3. The generation process of the LISA-Mask dataset.

Figure 3 exhibited the generation process of LISA-Mask dataset. First, YOLOv2 is trained on LISA training set and named as $Model_0$, then 50 different masks are generated by using the aforementioned method, and then the attack on $Model_0$ is performed on different masks based on the method in [8], respectively, the difference of the detection results with the true labels is added to the loss function, and the attack patches are

updated by back-propagation training. The generated patches are applied on LISA, and the images with the patch attacks are obtained, that is named as LISA-Mask dataset. The produced dataset contains a total of 11 categories of traffic sign images, each contained 3000 images that were attacked 50 times, for a total of 33,000 images.

3 Experiments and Results

3.1 Test Bench Setup

To evaluate our proposed work, we constructed the experimental data according to the structure in Fig. 4. Firstly, the LISA-Mask and LISA data sets are merged. There are 11 types of targets and each type of target is divided into clean data and attacked data. Then, to keep data balance in training, three enhancement methods is used on categories less than 100 pictures in the LISA dataset: contrast, brightness and sharpness change. We don't recommend using cutting, mirroring, rotation and other enhancement methods, for these complex situations are not common in driving detection task. Finally, we selected two hundred images randomly from each category of data to construct the experimental dataset, which is split into 80% training and 20% test set.

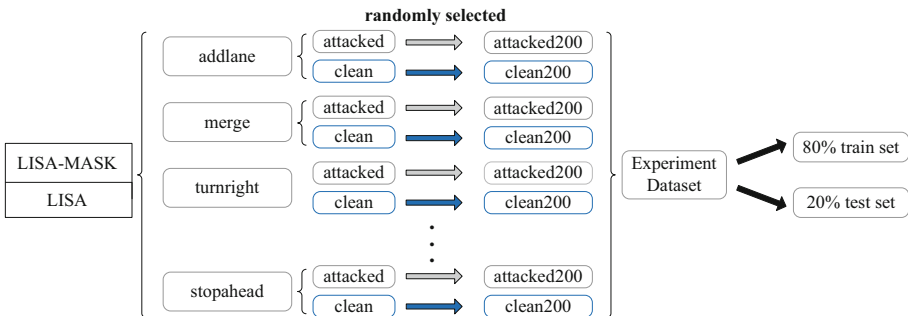


Fig. 4. Construction structure of the experimental dataset.

For all experiment, we use tensorflow1.14 and P4000 for training. YOLO is trained by Adam optimizer with learning rate 0.01, and batch size is 32. In the training of adversarial patches, SGD is used with learning rate 0.01, and decay rate is set to 0.1.

3.2 Object Detection

Object Selection Performance Analysis of IYOLO-TS on Clean Dataset

To evaluate the performance of IYOLO-TS, we calculate the AP of YOLOv2 and IYOLO-TS for each class on the LISA test set in Table 1. It can be observed that IYOLO-TS has less reduced in AP for each class compared to YOLOv2. On average, the mAP of IYOLO-TS is 97.75%, which is only 1.25% lower compared to YOLOv2, indicating that IYOLO-TS can maintain a strong roadmap detection.

Table 1. Performance of YOLOv2 and IYOLO-TS on the LISA test set

Classes	addlane	keepright	laneend	merge	signalahead	limit25
YOLOv2	100%	100%	92.39%	98.53%	100%	100%
IYOLO-TS	100%	100%	91.76%	96.49%	100%	98.63%
	limit30	limit45	stopahead	stop	turnright	mAP
	100%	100%	100%	100%	100%	99.00%
	100%	100%	100%	100%	100%	97.75%

Analysis of the Validity of IYOLO-TS Defense Detection

To evaluate the defensive capability of IYOLO-TS, we calculated AP of each class on the dataset. It can be seen that IYOLO-TS can distinguish the adversarial samples from the clean data, and the mAP reaches 98.12%. Table 2 shows the detection AP of IYOLO-TS for all classes of images, and it can be seen that IYOLO-TS has a strong defense detection performance. Figure 5 shows the performance of IYOLO-TS and YOLOv2 against patch attacks. As can be seen that, compared to YOLOv2, IYOLO-TS achieves higher metrics in all the other 10 classes of flags except the signalahead class, which shows a stronger defense against attacked data.

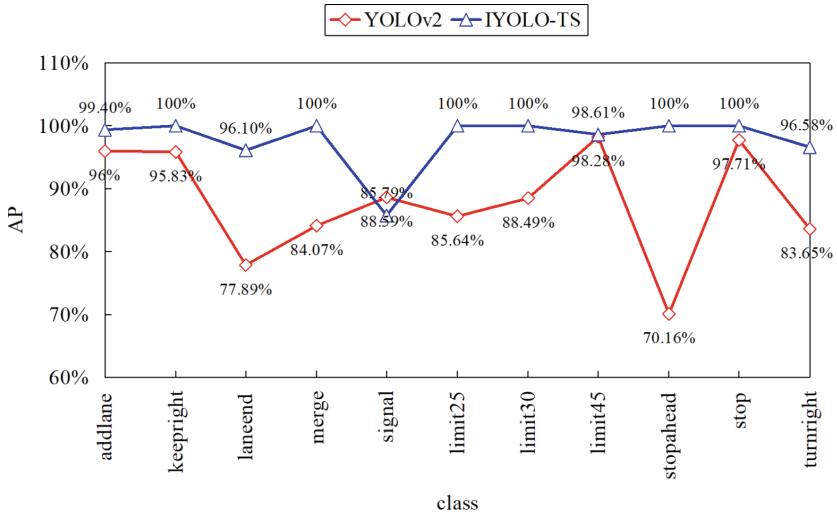
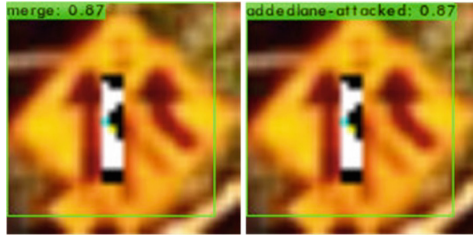


Fig. 5. Performance comparison of IYOLO-TS and YOLOv2 against patch attacks.

Figure 6 shows the defense effect on LISA-Mask. The attacked addedlane is able to successfully trick YOLOv2 to identify it as the merge class, however, IYOLO-TS is able to successfully and correctly identify the attacked target.

Table 2. IYOLO-TS AP for 22 classes of images, where classes indicate clean traffic signs data and classes-ad indicate attacked traffic signs data

Classes	AP	Classes-ad	AP
addlane	96.34%	addlane-ad	100%
keepright	100%	keepright-ad	100%
laneend	100%	laneend-ad	100%
merge	91.46%	merge-ad	97.67%
signalahead	92.5%	signalahead-ad	93.51%
limit25	100%	limit25-ad	100%
limit30	100%	limit30-ad	100%
limit45	95.74%	limit45-ad	95.65%
stopahead	100%	stopahead-ad	100%
stop	100%	stop-ad	95.45%
turnright	97.37%	turnright-ad	100%
mAP	98.12%		

**Fig. 6.** Performance of YOLOv2 and IYOLO-TS for detection of attacked added lane.

In addition, IYOLO-TS adds 11 additional attacked classes to the structure of YOLOv2, as Fig. 7 shows the detection results of some of the attacked classes. It can be seen that IYOLO-TS is not only able to correctly identify the attacked traffic sign, but also distinguish whether the traffic sign is under attack or not. It shows that IYOLO-TS has good detection ability for different kinds of patch attacks.

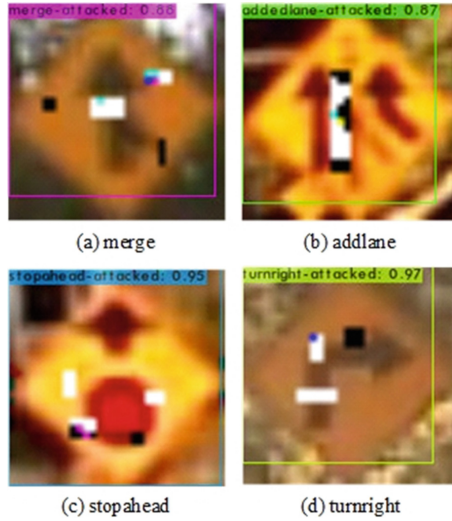


Fig. 7. Detection results of partially attacked classes.

3.3 Analysis of the Effectiveness of Patch Attack Defense

In order to evaluate the defensive capability of IYOLO-TS, we test IYOLO-TS under white-box attacks and physical world attacks respectively.

Defense Effectiveness Analysis under White-box Attacks

We continue with the LISA-Mask generation process, by using RP_2 to generate the patch dataset $LISA-Mask_0$ against IYOLO-TS. First, IYOLO-TS was trained on the LISA training set, and then images with the patch attack were generated on the LISA dataset using RP_2 against the trained IYOLO-TS to obtain the $LISA-Mask_0$ dataset. Then, the generated patch dataset $LISA-Mask_0$ was used to test the IYOLO-TS model. Table 3 shows the performance of IYOLO-TS against white-box attacks.

Table 3. Detection effectiveness of IYOLO-TS against white-box attacks

Classes	addlane	keepright	laneend	merge	signalahead	limit25
AP	95.95%	100%	90.79%	90.24%	100%	100%
	limit30	limit45	Stopahead	stop	turnright	mAP
	94.37%	95.35%	98.61%	100%	100%	97.22%

As can be seen from the Table 3, except for laneend and merge, which have an accuracy of about 90%, other classes have AP values higher than 94%, indicating that IYOLO-TS still shows a strong defense capability in the face of new attacks.

Defense Effectiveness Analysis under Physical World Attacks

To verify the usefulness of the model in this paper, the defensive performance of IYOLO-TS in the physical world was tested. In the experiments, the generated adversarial patches are printed and attached to the traffic signs to further compare and demonstrate the defense effectiveness of YOLOv2 and IYOLO-TS. As shown in (a) (d) (g) (j) of Fig. 8, YOLOv2 miscalculates under the generated adversarial patch, and the performance of (b) (c) (e) (f) (h) (i) (k) (l) shows that IYOLO-TS can distinguish the clean data from the attack data under physical attacks.

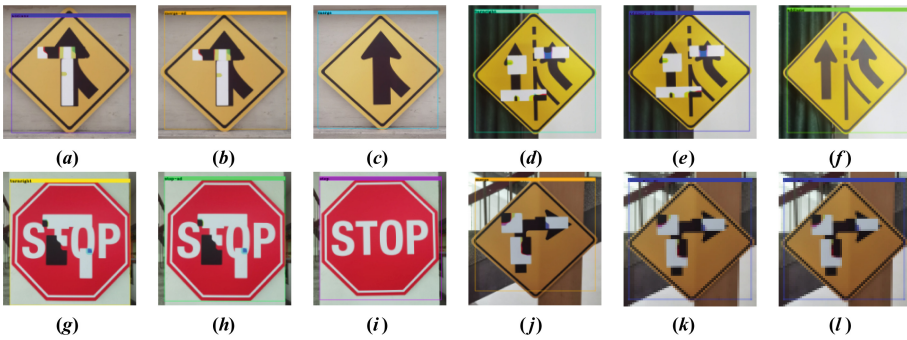


Fig. 8. Physical world attack test sample.

4 Conclusion and Future Work

In this paper, an improved defense model, IYOLO-TS, was firstly proposed to improve the anti-attack ability of the traffic sign detection. Firstly, the masks under multi-scale and multi-constraint conditions were built to simulate random multi-type physical attacks in the physical world, and the first test data set, Lisa-Mask is constructed through annotation fusion. On this basis, 11 attacked classes are innovatively added to the YOLOv2 network structure, so that the model can distinguish the attack samples from the original samples while maintaining the detection capability. In the experiment, we compared the detection performance of IYOLO-TS and YOLOv2, and completed the performance test and analysis of white-box attack and physical world attack respectively. Experimental results show that IYOLO-TS has a good defense ability against the adversarial patch attack from the physical world. But it can also be found that the real road traffic signs obscured, to be damaged, is far beyond this study at this stage can simulate. In addition, vehicle speed, weather, light and other factors will directly affect the processing efficiency of the model. Therefore, in our next work, how to optimize the model to adapt dynamic environment and achieve a more accurate and interpretable detection method are also important and interesting research topics.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 62106060).

References

1. Balasubramaniam, A., Pasricha, S.: Object Detection in Autonomous Vehicles: Status and Open Challenges. arXiv preprint [arXiv:2201.07706](https://arxiv.org/abs/2201.07706) (2022)
2. Salah Zaki, P., Magdy William, M., Karam Soliman, B., Gamal Alexsan, K., Khalil, K., El-Moursy, M.: Traffic Signs Detection and Recognition System using Deep Learning. arXiv preprint [arXiv:2003.03256](https://arxiv.org/abs/2003.03256) (2020)
3. Yi Huang, Wai-Kin Kong A.: Transferable Adversarial Attack based on Integrated Gradients. arXiv preprint [arXiv:2205.13152](https://arxiv.org/abs/2205.13152) (2022)
4. Cilloni, T., Walter, C., Fleming, C.: Focused Adversarial Attacks. arXiv preprint [arXiv:2205.09624](https://arxiv.org/abs/2205.09624) (2022)
5. Mo, Z., Patel, V.M.: On Trace of PGD-Like Adversarial Attacks. arXiv preprint [arXiv:2205.09586](https://arxiv.org/abs/2205.09586) (2022)
6. Subramanya, A., Pillai, V., Pirsiavash, H.: Fooling Network Interpretation in Image Classification. arXiv preprint [arXiv:1812.02843](https://arxiv.org/abs/1812.02843) (2019)
7. Singh, I., Araki, T., Kakizaki, R.K.: Powerful Physical Adversarial Examples Against Practical Face Recognition Systems. arXiv preprint [arXiv:2203.15498](https://arxiv.org/abs/2203.15498) (2022)
8. Eykholt, K., et al.: Robust physical-world attacks on deep learning visual classification. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1625–1634 (2018)
9. Thys, S., Ranst, W., Goedeme, T.: Fooling automated surveillance cameras: adversarial patches to attack person detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 49–55 (2019)
10. Lee, M., Zico Kolter, J.: On physical adversarial patches for object detection. arXiv preprint [arXiv:1906.11897](https://arxiv.org/abs/1906.11897) (2019)
11. Hayes, J.: On visible adversarial perturbations & digital watermarking. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Work-shops (CVPRW), pp. 1597–1604 (2018)
12. Naseer, M., Khan, S., Porikli, F.: Local gradients smoothing: defense against localized adversarial attacks. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1300–1307 (2019)
13. McCoyd, M., et al.: Minority reports defense: defending against adversarial patches. arXiv preprint [arXiv:2004.13799](https://arxiv.org/abs/2004.13799) (2020)
14. Wang, J., Chen, Y., Gao, M., Dong, Z.: Improved YOLOv5 network for real-time multi-scale traffic sign detection. arXiv preprint [arXiv:2112.08782](https://arxiv.org/abs/2112.08782) (2021)
15. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. CoRR (2016)
16. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
17. Bochkovskiy, A., Wang, C.Y., Liao, H.: YOLOv4: Optimal Speed and Accuracy of Object Detection (2020)
18. Ge, Z., Liu, S., Wang, F., et al.: YOLOX: Exceeding YOLO Series in 2021 (2021)
19. Levering, A., Tomko, M., Tuia, D., Khoshelham, K.: Detecting Unsigned Physical Road Incidents from Driver-View Images. arXiv preprint [arXiv:2004.11824](https://arxiv.org/abs/2004.11824) (2020)
20. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1528–1540 (2016)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Threat Intelligence



Research on Named Entity Recognition Method of Network Threat Intelligence

Keke Zhang¹, Xu Chen¹(✉), Yongjun Jing¹, Shuyang Wang¹, and Lijun Tang²

¹ North Minzu University, Yinchuan Ningxia 750021, China
chenxu@nmu.edu.cn

² Ningxia University, Yinchuan Ningxia 750021, China

Abstract. With the continuous emergence of new network threat means, how to turn passive defense into active prediction, the rise of Cyber Threat Intelligence (CTI) technology provides a new idea. CTI technology can timely and effectively obtain all kinds of network security threat intelligence information to help security personnel quickly identify all kinds of attacks and make effective decisions in time. However, there are not only a large number of redundant information in threat intelligence information, but also the problems of Chinese English mixing, fuzzy boundary, and polysemy of related security entities. Therefore, identifying complex and valuable information from this information has become a great challenge. Through the research on the above problems, a named entity recognition model in the field of Network Threat Intelligence Based on BERT-BiLSTM-Self-Attention-CRF is proposed to identify the complex network threat intelligence entities in the text. Firstly, the dynamic word vector is obtained through Bert to fully represent the semantic information and solve the problem of polysemy of a word. Then the obtained word vector is used as the input of BiLSTM, and the context feature vector is obtained by BiLSTM. Then the output result is introduced into the self-attention mechanism to capture the correlation within the data or features, and finally the result is input into CRF for annotation. To verify the effectiveness of the model, experiments are carried out on the constructed network threat intelligence data set. The results show that the model significantly improves the effect of Threat Intelligence named entity recognition compared with several other classical models.

Keywords: Cybersecurity · Named entity recognition · BERT

1 Introduction

With the acceleration of the world's digitization process, the network environment is becoming more and more complex. At the same time, the network attack behavior tends to be industrialized, and the attack means are becoming more and more diversified. The traditional way of building defense strategies and deploying products based on experience is difficult to detect [1], intercept, analyze and respond in time and effectively in the face of emerging new, persistent, and advanced threats [2]. In this context, Cyber Threat Intelligence (CTI) [3] technology came into being. As an important network

security knowledge, it can support the construction of a more active network security defense [4] mode. Based on all-around intelligence perception and multi-dimensional fusion analysis, it can study and judge the overall situation of network security and reasonably predict the threat trend, so as to realize dynamic and accurate response to network security threats. However, the existing network threat intelligence information is also mixed with a large number of invalid or interference information. How to more effectively obtain more critical threat intelligence entity information (such as organization, software, vulnerability number, etc.) from threat intelligence has become the focus of current research. Applying named entity recognition (NER) technology [4] to the field of Network Threat Intelligence can effectively solve the problem of extracting important security entity information from unstructured Threat Intelligence text. Automatically identifying network security entities from Internet information, such as software, vulnerabilities, attack means, and related network terms, and classifying them is an important step in constructing the knowledge map in network security [5].

2 Relation Work

In the early stage, NER tasks were performed using a rule-based and dictionary-based approach, which achieved good results when formulating very comprehensive rules and dictionaries, but at great cost, so machine learning methods were considered to improve the accuracy of NER. Mulwad V et al. [6] identified potential vulnerability descriptions through an SVM classifier and used Wikilog knowledge base to identify vulnerabilities, threats, and attacks in Web text. Since SVM cannot consider context information, Joshi A et al. [7] used CRF based system to identify important entities and concepts related to network security in a given text. In order to better improve the performance of NER, we can also consider adding POS, Weerawardhana S et al. [8] identified the key PAG parameters embedded in the vulnerability description text by machine learning and POS, including software name, version, impact, attacker operation, and user operation. It is proved by experiments that entity recognition tasks are carried out in the field of network security. The POS method does provide a viable alternative to machine learning.

Although machine learning [9] has some improvement on NER tasks in network security, it requires network security researchers to label security data, which is extremely costly. As a branch of machine learning, deep learning has become increasingly popular in recent years. At present, some researchers have applied deep learning to the field of named entity identification of network threat intelligence. Pingchuan Ma et al. [10] proposed a BiLSTM-CRF method to extract security-related concepts and entities from unstructured text and used open-source data to evaluate the model on P, R, and F1-score with good results. Wu H et al. [11] added a domain dictionary matching correction method based on BiLSTM-CRF, using BiLSTM to automatically capture context features, using CRF to learn label constraint rules, and using ontology domain dictionary to match correction. Qin Y et al. [12] added a feature template (FT) to BiLSTM-CRF to extract local context features, and CNN to extract character-level features of security entities, such as malware and English naming vulnerabilities. Li T et al. [13] proposed a neural network model based on self-attention to identify entities. On the basis of the existing BiLSTM-CRF model, the self-attention mechanism was added to extract more

context information related to the current word in a sentence and get more information about the current word. Han Zhang et al. [14] added GAN to BiLSTM-Attention-CRF to obtain tag data and solve the problem of lack of tag data in network security. P Evangelatos et al. [15] proposed using a transformer to extract named entities in threat intelligence and verified its validity by experimenting with the threat intelligence (DNRTI) dataset [16].

However, there is a polysemy in the named entity of Network Threat Intelligence. The word vectors obtained by word2vec and glove are static, which cannot solve the problem. At the same time, BiLSTM alone cannot obtain more information about the current word. Therefore, this paper proposes a BERT-BiLSTM-CRF named entity recognition method that combines a self-attention mechanism. BERT (Bidirectional Encoder Representations from Transformers) [17] the pre-training language model is a dynamic word vector based on the language model, which can dynamically adjust the embedding of words according to the semantics of the context, better express the representation relationship between words and sentences, and solve the problem of polysemy. In addition, the self-attention mechanism pays more attention to the important words related to the target entity in a sentence, which can better capture the interdependence between the current word and other words and extract more context information related to the current word.

3 BERT-BiLSTM-Self-attention-CRF Model

The BERT-BiLSTM-Self-attention-CRF model is divided into four parts: BERT pre-training language model, BiLSTM layer, Self-attention layer, and CRF layer. The unstructured text information is converted into dynamic word vectors through BERT, then the word vectors are used as input to BiLSTM. The context feature information is obtained from the forward LSTM and the reverse LSTM, and then some important information is selectively paid more attention and assigned higher weight through the

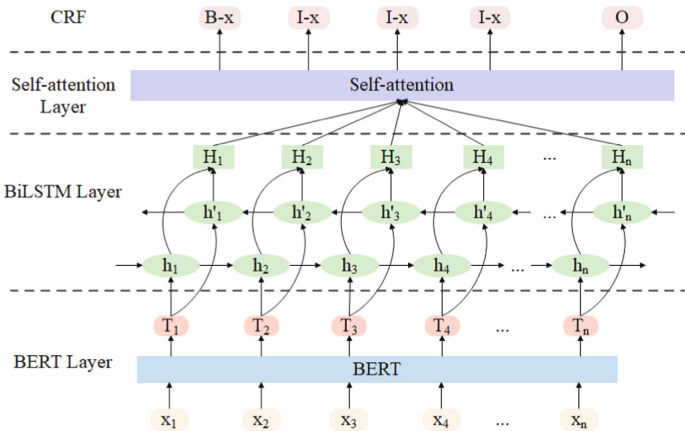


Fig. 1. BERT-BiLSTM-self-attention-CRF model architecture

self-attention mechanism. Finally, it is marked in the way of BIO through CRF. The model structure is shown in Fig. 1.

3.1 BERT Model

Language models are the most important part of named entity recognition, which transforms the input unstructured text into word vectors. Word2Vec [18] was originally used to get the word vector representation in the research of the named entity recognition of network threat intelligence. Its core idea is to obtain the vectorized representation of the word through the word context, including Skip-gram and CBOW. The former predicts the surrounding word by the given central word, and the latter predicts the central word by the given context information. In addition, the word vector representation is obtained by using the co-occurrence matrix with the Glove [19] method, which considers both local and global information. However, Word2vec and Glove are both static word vectors, and the word vector representation is the same in different contexts. For complex network security texts, there is a situation of polysemy. To solve this problem, this paper proposes a BERT pre-training language model, which can generate dynamic word vector representation to obtain the final representation of word vectors, so as to solve the problem of polysemy.

BERT adopts the encoding part of the bidirectional transformer and has two pre-training tasks. The first task is Mask Language, which randomly masks 15% of the words with MASK for the input text content, and then infers the masked words from the context information. The second task is to predict whether the second sentence is the next sentence of the first sentence, which is based on the first task, is marked with IsNext/NoNext by randomly selecting two sentences in the pre-training text. Figure 2 shows the structure of the BERT model.

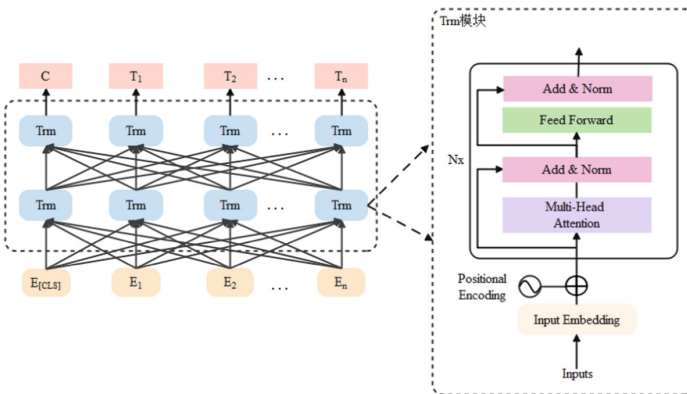


Fig. 2. BERT architecture

The input representation of BERT consists of three parts: Token Embedding, Segment Embedding, and Position Embedding. By adding and summing these three vectors together as the final input, feature extraction is performed in the encoding part of the

bidirectional transformer, and finally the sequence vector with rich semantics. The input representation is shown in Fig. 3.

Input	[CLS]	Cacti	has	an	authorization	issue	vulnerability	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{Cacti}	E_{has}	E_{an}	$E_{authorization}$	E_{issue}	$E_{vulnerability}$	$E_{[SEP]}$
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_B
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7

Fig. 3. Input representation of BERT

3.2 BiLSTM Layer

The traditional neural networks cannot memorize the input context information and infer the content from the previous information. This paper uses LSTM to solve this problem better. The model has a memory function, and can better capture the long-distance dependency. It can learn the information that needs to be forgotten and needs to be remembered through training. Its structure is shown in Fig. 4.

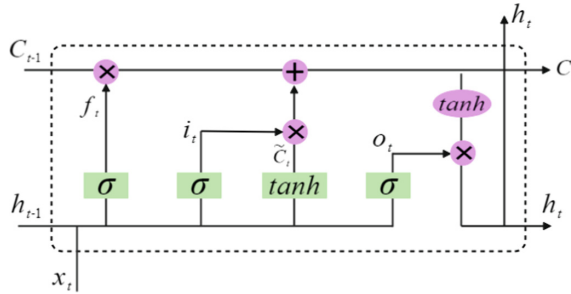


Fig. 4. LSTM structure

Its structure is composed of a forgetting gate, a memory gate and an output gate. It is controlled by the unit status. The implementation of LSTM is denoted as follows:

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

where x_t is the input vector, f_t is the forgetting gate, i_t is the memory gate, o_t is the output gate, C_t is the unit status of the time t , and h_t is the hidden state of the time t .

However, the LSTM cannot encode the information from the back to the front. Adding the reverse LSTM can better obtain the following information, that is, the BiLSTM model can better capture the bidirectional semantics, as shown in Fig. 5.

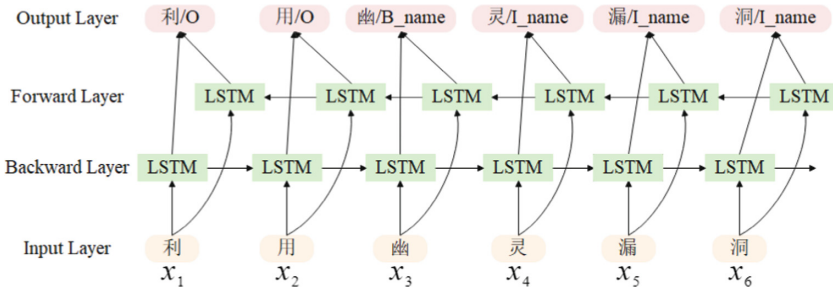


Fig. 5. BiLSTM model structure

In the text, the word vector output from the Bert layer is used as the input of the forward LSTM to obtain the forward feature information h_t and the reverse feature information h'_t , and then the two are spliced to obtain the final hidden state H_t , as shown below:

$$H_t = [h_t, h'_t] \tag{7}$$

3.3 Self-attention Layer

In order to better understand the effective information in the threat intelligence text, this paper proposes to add a self-attention mechanism after BiLSTM, which can capture the correlation between vectors, selectively pay more attention to some important information in the feature vector of BiLSTM layer output, give higher weight, and give lower weight to other information. The process of calculation the self-attention mechanism in this paper is as follows.

First, the hidden state of the BiLSTM layer output is represented as H_t , and the vector-matrix Q , K , and V are obtained by mapping the vector H_t :

$$\begin{aligned} Q &= H_t W^Q \\ K &= H_t W^K \\ V &= H_t W^V \end{aligned} \tag{8}$$

where W^Q , W^K , and W^V are the parameters learned in the training process, and then calculated by scaling the dot product attention. The calculation formula is as follows:

$$Attention(Q, K, V) = Soft \max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (9)$$

$1/\sqrt{d_k}$ is used to prevent the result from being too large. Finally, the result is normalized by using the Softmax function and multiplied by V to get the result.

3.4 CRF Layer

Conditional random fields (CRF) is a conditional probability model used to solve the maximization of sequence probability. In the threat intelligence NER task, BiLSTM is good at processing long-distance text information, but cannot deal with the dependency between adjacent tags. CRF can obtain the best prediction sequence through the relationship between adjacent tags, which makes up for the deficiency of BiLSTM. CRF ensures the validity of prediction tags by adding restriction rules to the final predicted tags. During the training process, these restriction rules are automatically learned by the CRF classifier, and the Viterbi is used to find the most likely tag sequence.

Given the input sequence $X = \{x_1, x_2, \dots, x_n\}$ of a sentence corresponds to the prediction sequence $Y = \{y_1, y_2, \dots, y_n\}$, and the score corresponding to the prediction sequence Y is calculated. The formula is as follows:

$$s(X, Y) = \sum_{i=0}^n A_{y_i, y_{i+1}} + \sum_{i=1}^n P_{i, y_i} \quad (10)$$

where A represents the transfer matrix of the label, P represents the label score, which is used to predict the probability of sequence Y , and the formula is as follows:

$$P(Y|X) = \frac{e^{s(X, Y)}}{\sum_{\tilde{Y} \in Y_X} s(X, \tilde{Y})} \quad (11)$$

where \tilde{Y} represents the correctly marked sequence and Y_X represents the marked sequence. Logarithmically on both sides of the above formula to obtain the likelihood function of the prediction sequence. The formula is as follows:

$$\ln(P(Y|X)) = s(X, Y) - \ln\left(\sum_{\tilde{Y} \in Y_X} s(X, \tilde{Y})\right) \quad (12)$$

Finally, a set of tag sequences with the highest probability is calculated by Viterbi.

4 Experimental Analysis

4.1 Dataset Construction

Since there is no public Chinese named entity identification dataset in network security, this paper mainly obtains the required data from the websites related to network security

vulnerability through python, such as the National Information Security Vulnerability Sharing Platform (www.cnvd.org.cn), Information Security Vulnerability Portal (<http://cve.scap.org.cn>) 360 Network Security Response Center (cert.360.cn) and national Internet Emergency Center (www.cert.org.cn) are divided into nine types (as shown in Table 1), labeled with BIO. B represents the first word of the entity, I represents the intermediate word of the entity, and O represents the non-entity.

Table 1. Entity labeled mode

Entity type	BIO mode
person	B-person/I-person
org	B-org/I-org
changjia	B-changjia/I-changjia
software	B-software/I-software
cve_id	B-cve_id/I-cve_id
cnvd_id	B-cnvd_id/I-cnvd_id
vul_name	B-vul_name/I-vul_name
date	B-date/I-date
term	B-term/I-term
other	B-other/I-other
non-entity	O

The labeled dataset is divided into the training set, test set, and verification set in 7:2:1 (as shown in Table 2).

Table 2. Dataset size

Type	Train	Test	Dev
person	1285	395	158
org	268	83	26
changjia	691	221	84
software	5639	1706	712
cve_id	1906	467	190
cnvd_id	1503	595	219

(continued)

Table 2. (continued)

Type	Train	Test	Dev
vul_name	3115	954	386
date	1250	484	200
term	734	191	65
other	1695	429	135

4.2 Evaluation Metrics

How to evaluate the performance of NER is a crucial step in the NER task. Through evaluation, we can analyze the advantages and existing problems of the proposed algorithm. At present, there are three main evaluation indicators to measure the performance of NER tasks: Precision, Recall, and F1-score.

Precision refers to the probability that all the samples predicted to be positive are actually positive. The formula is as follows:

$$P = \frac{TP}{TP + FP} * 100\% \quad (13)$$

For the original sample, the recall rate refers to the probability of being predicted as a positive sample in the actually positive sample. The formula is as follows:

$$R = \frac{TP}{TP + FN} * 100\% \quad (14)$$

Obviously, the above two evaluation indicators are contradictory and cannot meet the requirements that the precision and recall can reach the best. Therefore, the F1-score is balanced, and the precision and recall rate are considered to maximize the two as much as possible. As a comprehensive index to balance the impact of precision and recall, its formula is as follows:

$$F1 = \frac{2 * P * R}{P + R} * 100\% \quad (15)$$

where TP refers to the number of samples that are actually positive and predicted to be positive, FP refers to the number of samples that are actually negative and predicted to be positive, FN refers to the number of samples that are actually positive and predicted to be negative.

4.3 Experimental Results

Experiments are carried out on the constructed network security data set. In order to verify the rationality of the proposed model, the model is compared with several classical models in the named entity recognition task. The comparison results are shown in Table 3.

For the task of named entity recognition in network security, more features are needed for recognition, and the state of the current time should be related to the state

Table 3. Comparison of different models (%)

Models	P	R	F1
HMM	85.78	80.97	81.14
BiLSTM	88.07	81.78	83.17
BiLSTM-CRF	89.22	83.67	85.37
BERT-BiLSTM-CRF	90.21	91.90	91.04
BERT-BiLSTM-Self-attention-CRF	92.45	94.20	93.32

of the previous time and the next time, while the current state in HMM is only related to the previous state. From the experimental results, it can be seen that the F1 value of BiLSTM is higher than that of HMM. BiLSTM cannot learn the relationship between state sequences. After adding CRF, it can learn state sequences. Compare the BiLSTM-CRF model with BERT-BiLSTM-CRF, the experimental results show that because BERT can deeply extract the semantic information of network security text and fully reflect the polysemy of a word, the F1-score has been significantly improved.

Comparing the BERT-BiLSTM-CRF model with the BERT-BiLSTM-CRF model proposed in this paper, which combines the self-attention mechanism, the precision, the recall, and F1-score are improved. Due to the addition of the self-attention mechanism, the model is better at capturing the correlation between the data in the full text of network security by calculating the interaction between words, so that the F1-score of the model proposed in this paper is 2.28% more than the BERT-BiLSTM-CRF model. It has achieved good results in the task of network security named entity recognition.

5 Conclusion and Future Work

Threatening intelligence has gradually become one of the hot areas of network security. At present, government departments and network security enterprises pay more attention to the development of threatening intelligence, and the demand for threatening intelligence in all walks of life is growing. However, there are some problems in Network Threat Intelligence entities, such as ambiguous words, mixed Chinese and English, blurred boundary, etc. To solve these problems, this paper presents network security named entity recognition model based on BERT-BiLSTM-CRF, which combines a self-attention mechanism, uses a BERT pre-training language model to generate word vectors dynamically through two-way Transformer structure, mining syntax structure, and semantic information, and introduces a self-attention mechanism to calculate the correlation between words. Distance dependence can be better solved by assigning different weights to different words according to their degree of association. Experiments show that the model has a certain improvement in P, R, and F1-score, and has a good recognition effect. It can complete the actual network threat intelligence entity identification work and solve the difficulties of threat intelligence entity identification and the ambiguity of one word.

However, there is still much space to improve the task of identifying named entities for network threat intelligence. Because there are still a large number of unmarked network security corpora in a specific area, transfer learning can be considered in future research to solve the problem of lack of labeled data. The performance of identifying network threat intelligence entities can be further improved by expanding the size of the corpus.

Acknowledgement. The work is supported by Supported by the Fundamental Research Funds for the Central Universities, North Minzu University (2022PT_S04), and the Natural Science Foundation of Ningxia Province (No. 2020AAC03212), and the Innovation Projects for Graduate Students of North Minzu University (Project No. YCX21087).

References

1. Han, X., Li, C., Li, X., Lu, T.: Research on APT attack detection technology based on DenseNet convolutional neural network. In: 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI), pp. 440–448 (2021)
2. Pujol-Perich, D., Suárez-Varela, J., Cabellos-Aparicio, A., et al.: Unveiling the potential of graph neural networks for robust intrusion detection. *ACM SIGMETRICS Perf. Eval. Rev.* **49**(4), 111–117 (2021)
3. Schlette, D., Caselli, M., Pernul, G.: A comparative study on cyber threat intelligence: the security incident response perspective. *IEEE Commun. Surv. Tutor.* **23**(4), 2525–2556 (2021)
4. Nozza, D., Manchanda, P., Fersini, E., et al.: Learning to adapt with word embeddings: domain adaptation of named entity recognition systems. *Inf. Process. Manag.* **58**(3), 102537 (2021)
5. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(2), 494–514 (2022)
6. Mulwad, V., Li, W., Joshi, A., et al.: Extracting information about security vulnerabilities from web text. In: 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 257–260 (2011)
7. Joshi, A., Lal, R., Finin, T., et al.: Extracting cybersecurity-related linked data from tex. In: 2013 IEEE Seventh International Conference on Semantic Computing, pp. 252–259 (2013)
8. Weerawardhana, S., Mukherjee, S., Ray, I., Howe, A.: Automated extraction of vulnerability information for home computer security. In: Cuppens, F., Garcia-Alfaro, J., Zincir Heywood, N., Fong, P.W.L. (eds.) *FPS 2014. LNCS*, vol. 8930, pp. 356–366. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17040-4_24
9. Zhao, Q., Sun, J., Ren, H., Sun, G.: Machine-learning based TCP security action prediction. In: 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp. 1329–1333 (2020)
10. Ma, P., Jiang, B., Lu, Z., et al.: Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields. *Tsinghua Sci. Technol.* **26**(3), 259–265 (2020)
11. Wu, H., Li, X., Gao, Y.: An effective approach of named entity recognition for cyber threat intelligence. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1370–1374 (2020)

12. Qin, Y., Shen, G.-W., Zhao, W., Chen, Y.-P., Yu, M., Jin, X.: A network security entity recognition method based on feature template and CNN-BiLSTM-CRF. *Front. Inf. Technol. Electron. Eng.* **20**(6), 872–884 (2019). <https://doi.org/10.1631/FITEE.1800520>
13. Li, T., Guo, Y., Ju, A.: A self-attention-based approach for named entity recognition in cyber-security. In: 2019 15th International Conference on Computational Intelligence and Security (CIS), pp. 147–150 (2019)
14. Zhang, H., Guo, Y., Li, T.: Domain named entity recognition combining Gan and BiLSTM-Attention-CRF. *Comput. Res. Dev.* **56**(9), 8 (2019)
15. Evangelatos, P., et al.: Named entity recognition in cyber threat intelligence using transformer-based models. In: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 348–353 (2021)
16. Wang, X., et al.: DNRTI: a large-scale dataset for named entity recognition in threat intelligence. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1842–1848 (2020)
17. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding (2018)
18. Ren, F., Jiang, Z., Liu, J.: A bi-directional LSTM model with attention for malicious URL detection. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 300–305 (2019)
19. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Text Recognition



Research on the Recognition of Internet Buzzword Features Based on Transformer

Dawei Xu^{1,2(✉)}, Yijie She¹, Zhonghua Tan³, Ruiguang Li⁴, and Jian Zhao¹

¹ College of Cybersecurity, Changchun University, Changchun, China
xudw@ccu.edu.cn

² School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China

³ College of International Education, Hainan Normal University, Haikou, China

⁴ National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China

Abstract. Accurate identification of Internet buzzwords plays an important role in positive Internet opinion guidance. A Transformer-based Internet buzzword feature recognition system was designed to address this problem. The traditional way of crawling data has been improved, a real-time crawling module has been added, and an Internet buzzword corpus has been constructed by itself. The traditional way of crawling data has been improved, a real-time crawling module has been added, and an Internet buzzword corpus has been constructed by itself. Traditional machine learning models suffer from gradient disappearance and gradient explosion, the Transformer model, with its parallel computing and self-attentive mechanism, is a good solution to these problems, and its bi-directional connection allows the parameters of the context to be updated uniformly, thus allowing better aggregation of information and solving the problem of scattered contextual information. Transformation of the position-encoded part of the Transformer model starts with a relative position representation (RPR). It compensates for its inability to obtain relative location information. The experimental results show that the improved Transformer model can achieve an accuracy rate of 90.1%, a recall rate of 92.13%, and an F1 value of 91.16% in recognizing Internet buzzwords.

Keywords: Internet buzzwords · Transformer model · Relative position representation (RPR)

1 Introduction

With an Internet penetration rate of 73.0% as of December 2021 [1], Internet has become an essential part of people's lives. Internet has given the public more channels to express their ideas, and Internet buzzwords are the concentrated product of expressing ideas, but there are positive and negative Internet buzzwords, and while they express the ideas of Internet users, they may produce negative public opinion guidance. Therefore, accurate identification of Internet buzzwords plays an important role in the guidance of correct Internet opinion.

The system applies deep learning techniques to achieve recognition of Internet buzzwords. Deep learning techniques can extract, transform and combine features from the initial text to obtain a set of feature representations, and then input a prediction function to obtain the recognition results [2]. Deep learning is built around the implementation of three functional components: the embedding layer, the encoding layer, and the output layer, embedding layer convert words into feature vectors, the Encoding layer obtains textual contextual features, and the output layer acquires the rules between sequences and classifies their output [3]. Although RNN structures are widely used to process sequence-like time-stream data [4–6], they suffer from structural problems such as serial computation, gradient disappearance [7], and one-way construction. The contributions of applying the Transformer model for web buzzword feature recognition are as follows: (1) In the data crawling, the module of real-time crawling is added, which can obtain the data of Internet buzzwords more accurately and improve the problem that the traditional crawling data is too slow to update. (2) The current web buzzword dataset is scattered and sparse so the data collected through web crawling is used to build a dynamic web buzzword corpus on its own. (3) Traditional machine learning models suffer from the problem of gradient disappearance and gradient explosion. The Transformer model, with its parallel computing and self-attentiveness mechanism, solves these problems, and its bi-directional connection allows the parameters of the context to be updated uniformly, thus enabling better information aggregation and solving the problem of information dispersion in the context. (4) Improvements to the start position of the Transformer model, converting the encoding vector of the starting position to a relative position [8] representation (RPR), compensate for the necessity to introduce explicit location information at the location code.

2 Related Work

The existing literature on the identification of Internet buzzwords and Internet neologisms summarizes three types: rule-based methods, statistical-based methods, and methods based on a combination of statistics and rules.

The rule-based approach focuses on developing rules that share common features between words, words, and words, based on linguistic theory and knowledge, or on observing the rules and patterns of word formation through long-term study of the language, and then summarizing their properties and combining them with grammar. As the core of the rule-based approach to new word discovery is the construction of a knowledge base for the domain, a more specialized rule base needs to be created, and new words need to be discovered based on the degree of similar recognition in its rule base when carrying out online buzzword identification. The statistical-based approach improves on the drawbacks of the rule-based approach which uses extensive manual annotation, saving significant time and labor costs. Even though the statistical-based approach makes up for many of the shortcomings of the rule-based approach, experiments in the literature have shown that the statistical-based approach has a low recognition rate that does not allow for good recognition of words, while a fusion of the two can improve the recognition rate of Internet buzzwords. The literature [9] proposes a k th order algorithm for PMI, and experiments show that its accuracy is improved by about 28.79% over

PMI, and it is found that when the parameter k takes a value greater than or equal to 3, it can overcome the defects of the PMI method. The Transformer model is also based on a combination of statistical and rule-based methods and has been applied to Internet buzzwords to improve recognition rates.

3 Overall System Architecture

The Transformer deep learning model is applied to identify the features of Internet buzzwords, and the overall process of the system is shown in (see Fig. 1).

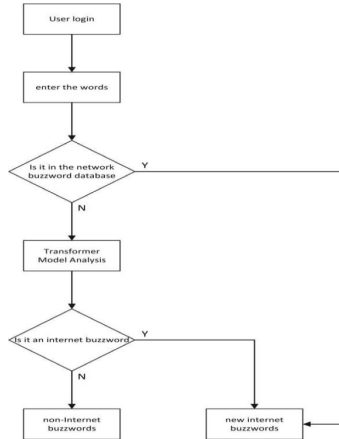


Fig. 1. Overall system flow chart

Firstly, the user logs in to the Internet buzzword recognition system and enters the text to be analyzed on the text analysis page. The Internet buzzword database in the background makes a judgment on the text entered, if it is an Internet buzzword in the corpus then it is directly identified as an Internet buzzword, if it does not exist in the Internet buzzword database then the input is entered into the Transformer model to determine if it is an Internet buzzword.

The Transformer Internet-based buzzword recognition technology solution is implemented in the following steps:

Step1, to crawl the existing Internet buzzword corpus on Weibo, to achieve real-time incremental crawling of Internet buzzwords on the original crawler technology, need to mark an identifier on the URL that is the data fingerprint, set the data fingerprint as a hash value, and then just compare the hash value to determine whether the crawled content needs to be updated.

Step2, the crawled Internet buzzwords were pre-processed by first de-duplicating the data, followed by word separation for the longer phrases, using search engine mode, and then filtering the deactivated words using Baidu’s deactivated word list.

Step3, use matplotlib library, jieba library, and word cloud library to realize the visual display of the processed Internet buzzwords and draw the word cloud of Internet buzzwords.

Step4, the pre-processed data is selected for the text vector representation by the Skip-gram method in the word2vec model.

Step5, for the feature vectors obtained in the previous step, position encoding is performed, and a position vector representing position information is combined on word embedding to obtain the final vector with position information.

Step6, input the vector with location information into the Transformer model and determine whether the input is a web buzzword or not.

4 System Implementation

4.1 A Subsection Sample

4.1.1 Data Acquisition

Real-time incremental crawling of Internet buzzwords is done by tagging URLs with a data fingerprint identifier. Set data fingerprint to the hash value, and generate a unique fixed-length string from the input words, the hash values are then compared to determine if the crawl needs to be updated. The former can insert a piece of data into the collection, returning 1 for success and 0 for failure; the latter can query whether an element exists in the collection, returning 1 for existence and 0 for non-existence. (see Fig. 2), when the Spider module receives a URL to process, a Spider middleware is added to determine whether the fingerprint of the URL exists in the Redis database and if so, the URL is discarded; if not, the new URL is fetched and crawled.

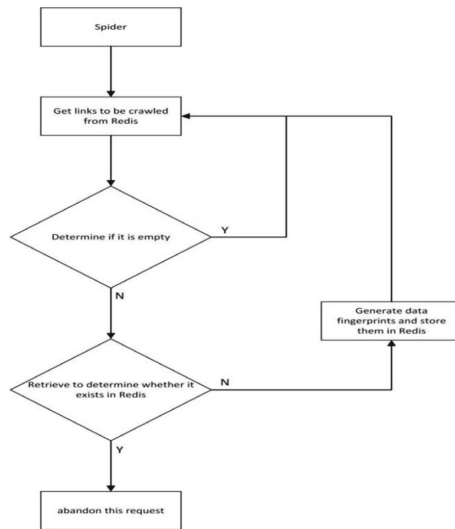


Fig. 2. Real-time web crawling flow chart

4.1.2 Data Pre-processing

By counting the content crawled by the keyword “Internet buzzwords”, a total of tens of thousands of high-frequency Internet buzzwords were crawled. Firstly, tens of thousands of buzzwords were de-duplicated, applying the duplicated() function of pandas, a data analysis tool in python, to detect duplicate data, duplicate rows with small indexes will return “True”, and data marked as True will need to be removed by applying the drop_duplicates() function.

The next step is to apply python’s third-party Chinese word splitting library, jieba, to the longer phrases in the crawled Internet buzzwords. According to the size of the granularity of the Internet, buzzword decided to use the more accurate search engine mode in the above for the word splitting process, for long words to cut the command as follows: jieba.cut_for_search(); jieba.lcut_for_search().

The next step is to filter the crawl data for English characters, numbers, mathematical characters, punctuation marks, single Chinese characters that are used very frequently, inflectional auxiliaries, adverbs, prepositions, conjunctions, etc. This article uses the Baidu deactivation word list filter.

4.1.3 Constructing an Online Buzzword Feature Vector

The pre-processed data is transformed into a character vector using the Word2vec model for characters. The Word2vec module is called from the Genism package. The Word2vec module contains two methods for vectorizing text, CBOW, and Skip-gram, respectively. In the training process, sg = 1 is set and the algorithm of Skip-gram is used for training. The window_size of the sliding window is set to 5, the dimension of the size word vector is set to 100, and min_count is used for the filtering operation. Words with a frequency less than the set value will be discarded, which is set to 5 in this paper. Skip-gram is the prediction of surrounding words using central words, for each central word there are K words as output, and there are K predictions for a word, for a total of $K * V$.

The model training process is as follows: (1) Use center_words V to query W0 and target_words T to query W1 to get two tensors of shape [batch_size, embedding_size], respectively, denoted as H1 and H2. (2) The two tensors are then dotted together. (3) Using a sigmoid function acting on (2), the result of the above dot product is normalized to a probability value of 0–1 as the predicted probability, and this model can be trained based on the label information L. After finishing the training of the model, W0 is generally used as the final word vector to be used, represented by a vector of W0. Using vector dot product, the similarity between different words can be calculated.

4.2 Transformer Model

The Transformer model was proposed by Vaswani A. et al. in their paper “Attention Is All You Need” [10], published in late 2017, and the general structure is shown in (see Fig. 3).

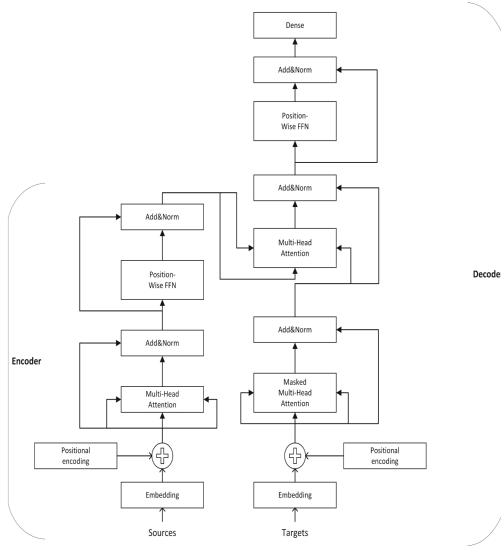


Fig. 3. Transformer structure diagram

An analysis of the location coding place in the traditional Transformer model, since the Transformer model, does not have the iterative operation of a recurrent neural network, and no access to relative position information, so the position information of each word must be provided to the Transformer. Transformation of the position encoding part of the Encoder, the decoder part of the Transformer model into a relative position representation (RPR), compensating for its inability to obtain relative location information.

Two-position encoding vectors of the model need to be learned, one for computing z_i and one for computing e_{ij} . If the middle index is k , then there will be $2k + 1$ relative position encoding vectors to learn, of which k are to its left, k is to its right, and one belongs to itself. Relative positional encoding is not used in the traditional Transformer to calculate the degree of attention i pays to j after SoftMax for word i and word j . Comparing the two calculation methods, it is easy to see that the RPR calculation is more accurate for the position, so the model uses RPR for both the Encoder and Decode parts of the position encoding.

5 Analysis and Visualization of Experimental Results

5.1 Experimental Parameters

The number of layers is set to 2 by default, and the value of 128 is set to True. BIDI-RECTIONAL is set to True to analyze the sequence from front to back and from back to front. Table 1 lists the parameters of the Transformer model and their corresponding optimal parameter values.

Table 1. Transformer model parameters.

Parameters	Value
BATCH_SIZE	128
HIDDEN_DIM	768
OUTPUT_DIM	1
N_LAYERS	2
BIDIRECTIONAL	True
DROPOUT	0.25
SEED	1234
MAX_INPUT_LENGTH	80
N_EPOCHS	5

5.2 Comparative Experiments

The experimental evaluation of the network structure for the recognition rate of Internet buzzwords was evaluated using the precision Pre, recall Rec, and F1 values to evaluate the effectiveness of Internet buzzword recognition. To verify the performance of the Transformer model proposed in this paper, the feature vectors of Internet buzzwords were used as the input vectors of the model, and the accuracy recognition results of the comparison experiments on top of the single models commonly used by CRF, LSTM, BILSTM and CNN [12] are shown in Table 2.

Table 2. Recognition performance of the models.

Model	Accuracy rate P (%)	Recall rate R (%)	F1 (%)
CRF	76.6	85.85	80.96
LSTM	20.83	24.01	22.3
BILSTM	87.46	87.6	87.53
CNN	63.47	67.41	65.38
TRANSFORMER	90.1	92.13	91.16

The experimental results show that the Transformer structure-based online buzzword recognition model is the best over the common single models of CRF, LSTM, BILSTM and CNN. The LSTM model has the lowest recognition rate for irregular words such as Internet buzzwords because it can only extract information from above, not below, and its F1 value is only 22.3% which is ineffective for the recognition of Internet buzzwords. The CNN model has an F1 value of 65.38%, which is an average performance in buzzword recognition compared to other models. The F1 value of the model using BILSTM is 87.53%, which is a 6.57% improvement compared to the CRF model and still performs

relatively well in buzzword recognition. Applying the Transformer model performed best in terms of precision Pre, recall Rec, and F1 values, with 90.1%, 92.13%, and 91.16% respectively.

The evolution of the experimental evaluation parameter accuracy P is shown in (see Fig. 4), the evolution of the evaluation parameter recall R is shown in (see Fig. 5), and the evolution of the evaluation parameter F1 is shown in (see Fig. 6).

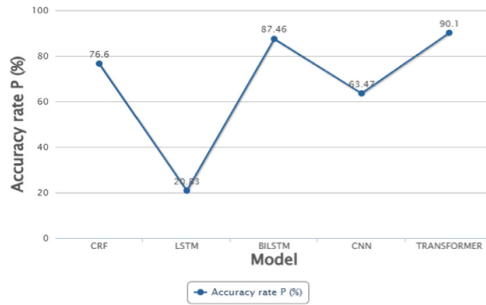


Fig. 4. Comparison of accuracy P (%) across models

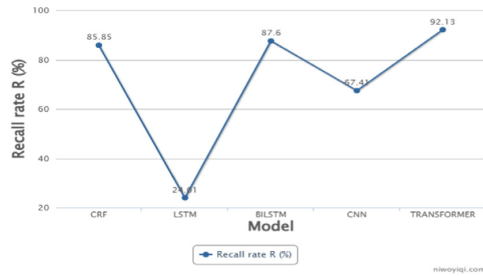


Fig. 5. Comparison of recall R (%) across models

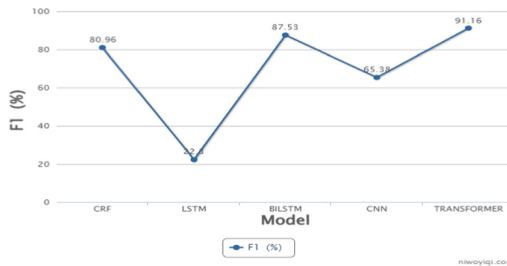


Fig. 6. Comparison of F1 (%) across models

The line graph of the experimental results reveals that the Transformer-based model has the highest accuracy, recall and F1 score, with the change curve at the top, at 90.1%, 92.13% and 91.16% respectively, and the experimental data shows that the model in this paper improves the recognition rate of Internet buzzwords.

6 Conclusion

To improve the recognition rate of Internet buzzwords, Transformer-based Internet buzzword feature recognition is proposed. The module of real-time crawling has been added to the data crawling, which can obtain the data of Internet buzzwords more accurately and improve the problem of too slow an update of traditional crawling data. As buzzword datasets on the web are scattered and sparse, a dynamic corpus of Internet buzzwords is constructed in-house from data collected through web crawling. Traditional machine learning models suffer from the problem of gradient disappearance and gradient explosion. The Transformer model, with its parallel computing and self-attentiveness mechanism, solves these problems, and its bi-directional connection allows the parameters of the context to be updated uniformly, thus enabling better information aggregation and solving the problem of information dispersion in the context. Improvements to the start position of the Transformer model, converting the starting position-coding vector to a relative position representation (RPR). It compensates for the need to introduce explicit location information at its location code.

Acknowledgments. This research was supported by the scientific research project of the Education Department of Jilin Province [NO. JJKH20220602KJ].

References

1. The 49th Statistical Report on the Development of the Internet in China. China Internet Network Information Center (2022)
2. Qiu, X.P.: *Neural Networks and Deep Learning*. China Machine Press, Beijing (2020)
3. Li, J., Sun, A., Han, J., et al.: A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **34**, 50–70 (2020)
4. Hammerton, J.: Named entity recognition with long short-term memory. In: North American Chapter of the Association for Computational Linguistics, pp. 172–175 (2003)
5. Huang, Z., Xu, W., Yu, K., et al.: Bidirectional LSTM-CRF models for sequence tagging. *Comput. Lang.* (2015)
6. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv: Learning* (2016)
7. Bengio, Y., Simard, P.Y., Frasconi, P., et al.: Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5**(2), 157–166 (1994)
8. Dai, Z., Yang, Z., Yang, Y., et al.: Transformer-XL: attentive language models beyond a fixed-length context (2019)
9. Du, L.P., Li, X.P., Yu, G., Liu, C.L., Liu, R.: New word discovery based on mutual information improvement algorithm for Chinese word separation system improvement. *Beijing Univ. J.* **52**(01), 35–40 (2016)
10. Vaswani, A., Shazier, N., Parmar, N., et al.: Attention is all you need. *Neural Inf. Process. Syst.* **30** (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Author Index

- Bao, Heng 73, 151
- Cai, Nishui 3
- Chen, Qian 28
- Chen, Xu 213
- Chen, Xunxun 151
- Cui, Jianming 199
- Deng, Jincheng 86, 113
- Deng, Lirui 73, 151
- Deng, Zhuxiang 3
- Du, Lin 177
- Fan, Xiangyu 113
- Gao, Jiaqi 129
- Guan, Jiazhi 151
- He, Yongqiang 51
- Huang, Jing 38, 187
- Jing, Yongjun 213
- Lan, Shizhan 38, 187
- Li, Ruiguang 129, 227
- Lin, Honggang 162
- Lin, Shenwen 139
- Liu, Fangming 86, 113
- Liu, Ming 199
- Long, Yangyu 86
- Ma, Lifang 187
- Mao, Hongliang 139
- Mei, Rui 51
- Qin, Yi 99
- She, Yijie 227
- Tan, Zhonghua 227
- Tang, Lijun 213
- Wang, Hao 3
- Wang, Qinqin 51
- Wang, Shuyang 213
- Wang, Ying 28, 99
- Wen, Weiping 51
- Wu, Fudong 129
- Wu, Zhen 139
- Xu, Chuanqi 177
- Xu, Dawei 129, 227
- Yan, Han-Bing 51
- Yang, Fen 28
- Yang, Jilong 86, 113
- Yang, Jinglin 139
- Yang, Peng 162
- Yang, Xueqin 162
- Zhai, Zhijia 28
- Zhang, Keke 213
- Zhang, Liang 151
- Zhang, Lin 99
- Zhang, Xing 28, 99
- Zhang, Yanjing 199
- Zhao, Jian 227
- Zhao, Wei 86, 113
- Zhao, Youjian 73
- Zhao, Zhangjie 99
- Zhu, Jiawei 129
- Zhu, Liehuang 129
- Zhu, Shengqiang 51