Anomaly detection using intelligent analytics

Chen, Zhaomin

2018

Chen, Z. (2018). Anomaly detection using intelligent analytics. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/82275>

<https://doi.org/10.32657/10220/46633>

Anomaly Detection using Intelligent Analytics

Chen Zhaomin

School of Computer Science and Engineering

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2018

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

2018.11.12

.....
Date

陈肇敏

.....
Chen Zhaomin

*To Dad and Mom,
To my Fiancée,
for their encouragements and love.*

Abstract

Anomaly detection is an important issue which has been studied in diverse application domains. In computer networks, the anomalies could indicate possible attacks to the target server. Thus, it is necessary to establish fast detection of network anomalies in the server for the purpose of intrusion detection in the server side. In addition, in the bottleneck router, it is also necessary to detect and mitigate the Low-rate Denial of Service attacks. In the field of computer vision, anomaly detection is often referred to as outlier detection, which is also very important, especially for the deep learning techniques. Nowadays, deep learning has become a mainstream approach in many research pertaining to computer vision. Deep learning technique requires large datasets. Thus, outlier detection techniques are often applied here to clean the datasets. In this dissertation, the author focuses mainly on developing novel network anomaly detection and image outlier detection techniques.

First, the author proposes Multi-scale Principal Component Analysis (MSPCA) algorithm to detect network anomalies. By combining PCA with wavelet analysis, MSPCA algorithm is supposed to address the normal subspace contamination problem so as to increase the detection accuracy. In addition, the author introduces MMECE threshold and Bayesian-PCA to the conventional-MSPCA algorithm. Thus, this improved-MSPCA algorithm has lower time complexity and fewer parameters to set. Furthermore, the author proposes a novel network anomaly detection system which combines the improved-MSPCA algorithm with sketch data structure. Upon the detection, it can identify the anomalous source IP addresses. As the proposed system detects the anomalies based on flow-level features, it has a better detection performance, compared to other packet-based detection systems.

Second, the author proposes Power Spectrum Entropy (PSE) to detect low-rate denial of service (LDoS) attacks in the bottleneck routers. PSE combines signal processing techniques with information entropy. It first decomposes the traffic signals into the frequency domain. Then the entropy of the power spectrum density is calculated. Due to the periodic property of the LDoS attack signal, its corresponding PSE value is comparatively smaller than that of the normal TCP signal. Therefore, PSE can be applied to detect the LDoS attacks. Furthermore, the author embeds PSE into the Robust RED queuing algorithm to propose the PSE-based Robust RED (PRRED) LDoS mitigation algorithm. PRRED consists of two detection phases: arrival time based detection and PSE-based detection. Arrival time based detection phase is meant to detect the suspicious set of incoming packets. PSE-based detection further tests whether these suspicious packets are LDoS attacks or not, based on the calculation of PSE. Once the packet is determined to be an attack packet, PRRED will drop this packet. These two detection phases help PRRED algorithm detect and mitigate the LDoS attacks efficiently and swiftly.

The author next focuses on outlier detection in the high-dimensional image datasets. Existing outlier detection works are not suitable for the image datasets due to the high dimension. The author combines autoencoder with Adaboost algorithm to address this issue. In this Adaboost-Autoencoder (ADAE) algorithm, multiple weak autoencoders will be optimized based on training data and the corresponding learning weights. ADAE ensembles a sequence of weak autoencoder to fully capture the feature correlations of normal images, so that it can achieve better detection performance compared to the traditional detection techniques. The author also introduces the Sparse Group Lasso (SGL) constraint to the optimization objective of each autoencoder, so that the compact detection model can be obtained. In order to optimize the SGL regularized learning objective, the author combines Adagrad optimization algorithm with Proxi-

mal Gradient Descent (Ada-PGD). Furthermore, Evolutionary Multi-Objective (EMO) based optimization framework is proposed here to determine the optimal penalty factors of SGL for each autoencoder. The compact detection model derived proves to be very efficient in outlier detection.

Acknowledgements

First I would like to express my sincere gratitude to my supervisor, Associate Professor Yeo Chai Kiat, my co-supervisor, Associate Professor Lee Bu Sung, and Associate Professor Lau Chiew Tong for the continuous guidance and encouragement during the four years of my Ph.D study. Thanks to their immense knowledge and patient instruction, I could learn a lot and get motivated in my research. Moreover, they offer various opportunities for me to broaden my academic experiences. This dissertation would not be possible without their invaluable support. I also want to give special thanks to Professor Yaochu Jin in the Nature Inspired Computing and Engineering (NICE) group in University of Surrey, UK.

I would like to take this opportunity to thank my labmates in Computer Network and Communication Graduate Lab (CNCL), including Yang Yiqun, Pham Thi Ngoc Diep, Zhang Yan, Chen Weiling, and Yean Seanglidet. They have always supported me with their great advices, companionship and encouragement.

My work cannot progress smoothly without the assistance of the lab officers. I would like to thank Mr. Toh Leong Teck for his hard work to maintain the lab and his help with technical issues.

Last but not least, I am thankful to my parents, my grandmother and my fiancée, for their continuous support and endless love. This thesis is dedicated to them.

Contents

Abstract	i
Acknowledgements	iv
List of Figures	viii
List of Tables	x
List of Abbreviations	xii
1 Introduction	1
1.1 Background	1
1.2 Network Anomaly Detection	3
1.2.1 Network Attacks	3
1.2.2 Principal Component Analysis based NIDS	6
1.3 Low Rate Denial of Service Attack	8
1.4 Image Outlier Detection	10
1.5 Contributions	12
1.5.1 Detection of Network Anomalies using Improved-MSPCA with Sketches	12
1.5.2 Detection and Mitigation of LDoS Attacks using Power Spectrum Entropy	13
1.5.3 Evolutionary Multi-objective Optimization of Ensemble Autoen- coder for Image Outlier Detection	14
1.6 Thesis Organization	15
2 Literature Review	16

2.1	Network Anomaly Detection	16
2.1.1	Signal Processing based Anomaly Detection	16
2.1.2	Statistical based Anomaly Detection	18
2.1.3	Machine Learning based Anomaly Detection	20
2.1.4	Unsupervised Learning based Detection	23
2.2	Detection of Low-Rate Denial of Service Attacks	25
2.2.1	Time Domain based Detection Methods	26
2.2.2	Frequency Domain based Detection Methods	28
2.2.3	Statistical based Detection	30
2.2.4	Other Detection Methods	31
2.3	Image Outlier Detection	32
3	Detection of Network Anomalies using MSPCA with Sketches	35
3.1	Conventional-MSPCA	37
3.2	Improved-MSPCA based Intrusion Detection Algorithm	39
3.2.1	Multiscale Multivariate Energy Contribution Efficiency (MMECE)	39
3.2.2	Bayesian PCA	41
3.2.3	Improved-MSPCA	44
3.2.4	Improved-MSPCA anomaly detection system	46
3.3	Evaluation of Improved MSPCA-based Anomaly Detection System . .	48
3.3.1	Detection Results of Improved-MSPCA	48
3.3.2	Comparing with Existing Intrusion Detection Methods	50
3.3.3	Conclusion on Improved-MSPCA algorithm	54
3.4	Sketch-MSPCA Anomaly Detection System	54
3.4.1	Sketch Data Structure	56
3.4.2	Detection System Architecture	57
3.5	Evaluation of Sketch-MSPCA system	59
3.5.1	MAWI Dataset	59
3.5.2	Evaluation of Different Sketch Sizes	61
3.5.3	Comparison with other Sketch-based Detection Systems	63
3.5.4	Anomaly Degrees for the Detected Attacks	68
3.6	Conclusion	69

4	Power Spectrum Entropy based Detection and Mitigation of Low-Rate DoS Attacks	71
4.1	Characteristics of Low-Rate DoS (LDoS) Attacks	72
4.1.1	Timeout-based LDoS Attacks	73
4.1.2	AIMD-based LDoS Attacks	74
4.2	Power Spectrum Entropy based Detection Metrics	76
4.2.1	Fourier Power Spectrum Entropy (FPSE)	76
4.2.2	Wavelet Power Spectrum Entropy	79
4.2.3	Modification of the two Metrics	82
4.3	Evaluation of Two Detection Metrics	83
4.3.1	Simulation Topology	83
4.3.2	Analysis of Two Information Metrics	84
4.4	Comparing with Existing LDoS Detection Methods	89
4.4.1	Comparing with Power Spectrum based Detection Systems	89
4.4.2	Comparing with other LDoS Detection Systems	94
4.5	PSE-based Robust RED (PRRED) Algorithm and its evaluations	96
4.5.1	Architecture of PRRED LDoS Mitigation Algorithm	97
4.5.2	Simulation Settings of Different Queuing Algorithms	100
4.5.3	Long-Lived TCP Traffics	101
4.5.4	Short-Lived TCP Traffic	103
4.5.5	Conclusion of PSE-based Robust RED algorithms	104
4.6	Conclusion	105
5	Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection	106
5.1	Adaboost-Autoencoder	107
5.1.1	General Autoencoder-based Outlier Detection	107
5.1.2	Combining Autoencoder with Adaboost	109
5.2	Multi-objective Optimization of Regularization with Sparse Group Lasso	110
5.2.1	Sparse Group Lasso	111
5.2.2	Adagra Proximal Gradient Descent (Ada-PGD)	113

5.2.3	Evolutionary Multi-objective Optimization of the Training Objective with SGL	114
5.2.4	Conclusion of the Detection Model	116
5.3	Evaluations	117
5.3.1	Image Datasets and Evaluation Metrics	117
5.3.2	Methods to be compared	118
5.3.3	Number of Iterations in ADAE	121
5.3.4	Evaluations of ADAE	121
5.3.5	Evaluation of ADAE with SGL constraint (ADAE-SGL)	124
5.4	Conclusions	126
6	Conclusions and Future Work	128
6.1	Conclusion	128
6.1.1	Detection of Network Anomalies using MSPCA with Sketches .	128
6.1.2	Power Spectrum Entropy based Detection and Mitigation of Low-Rate DoS Attacks	129
6.1.3	Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection	130
6.2	Future Research Directions	130
	Author's Publications	133
	Bibliography	134

List of Figures

1.1	Anomaly demonstration in a two-dimensional dataset	2
1.2	Low Rate Denial of Service attack targeting at the bottleneck routers .	9
1.3	Example of image outlier detection. The images consist of normal images ('Cat') and other outlier images	10
3.1	Procedures of Conventional-MSPCA	37
3.2	Procedures of Improved-MSPCA	44
3.3	Main Procedures of the Improved-MSPCA Detection System	46
3.4	ROC curve of Improved-MSPCA detection algorithm	49
3.5	Comparison of Improved-MSPCA algorithms for mother wavelet with different orders of vanishing moments	50
3.6	Scree plots for PCA and KL_PCA	51
3.7	ROC curves for PCA-based, KL-PCA-based and MSPCA-based detection algorithm	52
3.8	Normal Subspace Contamination of two algorithms	55
3.9	The updating procedures of sketch structure	56
3.10	Sketch-MSPCA Anomaly Detection System	57
3.11	ROC curves of different sketch numbers for the data on 2015.05.02 . . .	62
3.12	ROC curves of three detection systems for the data on 2015.05.02 . . .	64
4.1	Low-Rate DoS Attack	73
4.2	AIMD-based LDoS Attacks	75
4.3	Theoretical results of PSD for the periodic square wave	77
4.4	Discrete signals for the LDoS attack flow and its PSD estimates	78
4.5	Discrete signals for the normal TCP flow and its PSD estimates	78

4.6	Wavelet Energy for LDoS attacks and normal TCP flows	81
4.7	NS-3 bottleneck network topology	84
4.8	<i>FPSE</i> Distributions of Normal Flows and Attack Flows	85
4.9	<i>WPSE</i> Distributions of Normal Flows and Attack Flows	88
4.10	Detection Results for LDoS Attacks with Different Attack Periods . . .	91
4.11	Detection Results for LDoS Attacks with Different Attack Durations . .	91
4.12	Architecture of PSE-based Robust-RED	99
4.13	Goodput of Different Mitigation Algorithms under variations of LDoS attacks	102
4.14	Goodput of Different Mitigation Algorithms under variations of LDDoS attacks	103
4.15	Evaluation Results for Transmission of Short-lived TCP Traffic	104
5.1	Illustration of Different Sparse Regularizers The dash lines represent the connection with zero weights. The white circles denote the removed neurons.	113
5.2	Framework for Multi-objective Optimization of SGL Using NSGA-II . .	116
5.3	The Diagram of ADAE with Sparse Group Lasso Constraint	117
5.4	AUC scores and Coefficients for different number of autoencoders . . .	120
5.5	Precision-Recall Curves evaluated on STL-10 aircraft dataset	123
5.6	AUC score comparisons on each category in STL-10 dataset	124
5.7	F1 score comparisons on each category in STL-10 dataset	124
5.8	Pareto Front obtained for the "airplane" image data in STL-10 composed of 40 solutions	125

List of Tables

3.1	Energy Contribution of each level in traffic data	40
3.2	Execution Time Comparison	52
3.3	Detection Results of MSPCA, PCA and wavelet-based algorithm	53
3.4	Parameter Settings of Sketch-MSPCA System	59
3.5	Detail Information of Network Traces of three days in 2013	60
3.6	Detail Information of Network Traces of three days in 2014	60
3.7	Detail Information of Network Traces of three days in 2015	61
3.8	Parameters of three detection systems	64
3.9	Detection Results of Packet-PCA	66
3.10	Detection Results of Entropy-PCA	66
3.11	Detection Results of Flow-MSPCA	67
3.12	Types of attacks detected by three systems	67
3.13	Anomaly Degrees for the Selected Anomalous Source IP Addresses . . .	68
4.1	Distribution Distances for different Mother Wavelets	88
4.2	Comparisons of False Positive Rate by using MAWI Dataset	93
4.3	Comparison of Detection Results	96
4.4	Parameter Settings for Different Queuing Algorithms	100
5.1	Detail Information about all the Evaluation Datasets	118
5.2	Average AUC and F1 scores on STL-10 Dataset	122
5.3	Detection Results of Different Algorithms on Caltech-101, Scene_data and ImageNet. Evaluation metrics are averaged over all categories . . .	122
5.4	Detection Results of ADAE	126
5.5	Detection Results of ADAE-SGL	126

List of Abbreviations

ABBREVIATIONS

FULL EXPRESSIONS

ACK	Acknowledgment
ADAE	Adaboost-Autoencoder
ADAE-SGL	Adaboost-Autoencoder with Sparse Group Lasso constraint
AEWMA	Adaptive Exponentially Weighted Moving Average
Ada-PGD	Adagrad optimization algorithm with Proximal Gradient Descent
AIMD	Additive-Increase Multiplicative-Decrease
ARX	AutoRegressive eXogenous
CTS	Cumulative Traffic Spectrum
CUSUM	Cumulative Summation
CWT	Continuous Wavelet Transform
CWND	Congestion Window
DARPA	Defense Advanced Research Projects Agency
DBN	Deep Belief Networks
DCAE	Deep Convolutional Autoencoder
DDoS	Distributed Denial of Service
DoS	Denial of Service
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
EMO	Evolutionary Multi-Objective
EWMA	Exponentially Weighted Moving Average
FP	False Positive
FPR	False Positive Rate
FPSE	Fourier Power Spectrum Entropy
FRRED	FPSE based Robust Random Early Detection
FTP	File Transfer Protocol
GAN	Generative Adversarial Network
HAWK	Halting Anomaly with Weighted choKing

HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
KL	Kullback-Leibler
KL-PCA	Kullback-Leibler Principal Component Analysis
KNN	K Nearest Neighbors
LDoS	Low-rate Denial of Service
LOF	Local Outlier Factor
MAWI	Measurement and Analysis of the WIDE Internet
MF-DFA	Multifractal Detrended Fluctuation Analysis
MMECE	Multiscale Multivariate Energy Contribution Efficiency
MSABMS	Multiple Sampling Averaging based Missing Sampling
CPR	Congestion Participation Rate
MSPCA	Multi-scale Principal Component Analysis
NIDS	Network Intrusion Detection System
OCSVM	One Class Support Vector Machine
OD	Origin-Destination
PC	Principal Component
PCA	Principal Component Analysis
PRC	Partial Rank Correlation
PRRED	Power Spectrum Entropy based Robust Random Early Detection
PSE	Power Spectrum Entropy
PSD	Power Spectrum Density
RED	Random Early Detection
RED-PD	Random Early Detection Proportional Derivative
ROC	Receiver Operating Characteristic
RQA	Recurrence Quantification Analysis
RRED	Robust Random Early Detection
RSVM	Robust Support Vector Machine
RTO	Recovery time objective
RTT	Round-trip delay time
SGL	Sparse Group Lasso
SNMP	Simple Network Management Protocol
SOM	Self-Organizing Maps
SPE	Square Prediction Error
SRC	Spearman Rank Correlation
SRTT	Smoothed Round Trip Time

STL	Self-taught Learning
SVDD	Support Vector Data Description
SVM	Support Vector Machine
SYN	synchronize
TCM-KNN	Transductive Confidence Machine For K Nearest Neighbors
TCP	Transmission Control Protocol
TP	True Positive
TPR	True Positive Rate
UDP	User Datagram Protocol
WPSE	Wavelet Power Spectrum Entropy
WRRED	WPSE based Robust Random Early Detection

Chapter 1

Introduction

1.1 Background

Anomaly detection refers to the identification of data which do not conform to the normal behavior of the majority. Researchers often call these anomalous data as anomalies or outliers. Anomaly detection is a principal research focus and has been widely applied to various application domains, such as credit card fraud detection [21], manufacturing system fault detection [55], ecosystem disturbance detection [109], network intrusion detection [46] and image outlier detection [90].

Anomalies refer to data which do not conform to the common pattern of the majority. Figure 1.1 gives a simple illustration of anomalies in a two-dimensional dataset. There are two normal clusters, named $C1$ and $C2$. Apparently, $O1$ and $O2$ are two anomalous data which are isolated by the normal regions. Anomaly detection is important because anomalies in the datasets may result in serious problems. For example, in credit card transaction data, anomalies could mean credit card fraud or misuse of another's account number. In the computer network field, the anomalous traffic could

indicate that the network is under some sort of attack. In medical image dataset, the outliers could correlate with some disease status. Therefore, anomaly detection plays an important role in these application domains.

Figure 1.1: Anomaly demonstration in a two-dimensional dataset

In this dissertation, the author focuses mainly on the anomaly detection in computer network and outlier detection in image dataset. In the field of computer network, network attacks need to be detected efficiently. Therefore, anomaly based network intrusion detection system is significant for the network servers. There also exists one kind of special attacks which targets at the network bottleneck. Thus, it is also necessary to develop a fast detection and mitigation algorithm in the routers to limit the damage caused by this type of attack. Furthermore, anomaly/outlier detection can help to create a clean huge dataset, be it image or other data, which is essential for many deep learning techniques.

1.2 Network Anomaly Detection

1.2.1 Network Attacks

With the fast development and steady growth of Internet, the number of internet users has increased tenfold in the last two decades. People, companies and countries are now completely relying on their computer and network systems. A by-product of this increase is that the amount of network attacks is rapidly increasing. Attackers attempt to hack into the computer network to steal, alter, or destroy the data on the target machines. Without security measures and controls in the system, extensive attacks can cause massive damage to and loss of the data in a very short time.

Basically, there are two types of network attacks: active attacks and passive attacks.

Active Attacks.

An active attack is one where the intruder modifies the network packets during the communication or sends the forged data to the network system. There are several types of active attacks:

Denial-of-Service (DoS) Attack. The DoS attack attempts to make the targets unavailable to its intended users. The common method that DoS attackers adopt is to flood the targets with enormous requests so as to overload the systems. This can result in no response to legitimate requests. In the distributed denial of service (DDoS) attack, the flooding requests come from multiple sources, which makes it more difficult to detect. There are several kinds of DoS attacks:

- *TCP SYN flood.* In a SYN flood attack, the attacker utilizes a fake IP address to send repeated SYN packet requests to the target server. The server responds

to these requests by sending the SYN-ACK packets back. However, the server will not get the acknowledgments due to spoofed IP addresses of the request packets. Due to these incomplete three-way handshake processes, the resources of the victim server will be exhausted. Therefore, the normal TCP requests will not get accepted.

- *ICMP flood.* It is also known as Ping flood, which is a common DoS attack. The attackers overwhelm the target servers by adopting Internet Control Message Protocol (ICMP) echo requests, which are known as pings.
- *Smurf attack.* It is a DDoS attack which broadcasts ICMP packets with spoofed source IP to the whole computer network. Most devices on the network attempt to respond to the spoofed IP, but receive no reply. Due to this fact, the victims' computers are slowed down.

Probe Attacks. It is an attack which the attackers send client requests to the target, with the goal of determining the weakness of the target system and of exploiting vulnerability so as to compromise the system. It is very easy to generate this kind of attacks so that probe attacks occur frequently in the computer network. Port scan is one of the most common probe attacks. It is basically used to explore the vulnerability of the ports of the target. The client requests are sent to a wide range of port numbers of the target to check the active ports. Thus, a known vulnerability of the corresponding services can be exploited.

Masquerade Attacks. In a masquerade attack, the attacker pretends to be an authorized user of the target system so as to gain access to it. Once the attackers gain access, they may modify or delete data, and make changes to the network settings, which inflicts more damages to the computer network. The most common method that the attackers utilize is to steal the login IDs and passwords of the legitimate users.

Session Replay Attacks. They are also known as playback attacks or replay attacks. This kind of attack attempts to have a valid data transmission delayed or repeated. In order to establish this attack, the attacker attempts to intercept a session by stealing a legitimate user's session ID. One typical example of a replay attack is to replay the message from an authorized user. By retransmitting these valid data, the attackers could gain sufficient access to the computer network.

Message Modification Attack. In a message modification attack, an intruder can direct a message to a particular address by changing the packet header address. One common modification attack is website defacement.

Passive Attacks.

A passive attack is one where the intruder spies on the communication channel but does not change the message. In such an attack, the intruder does not try to hack into the system or change the communication data. Thus passive attack is very hard to detect as it does not cause damage to your system or change the information. There are two main kinds of passive attacks, namely, release of message contents and traffic analysis.

- *Release of message contents.* In this type of passive attack, the attack may monitor the message of some transmissions, like a telephonic conversation or an e-mail message.
- *Traffic analysis.* In this kind of attack, the intruder analyzes the network traffic data to get the frequency and length of the message exchange. Using all these information, they can get everything they want. Thus, during the process, all incoming and outgoing traffic is analyzed but not changed.

Although the network administrators already establish the firewalls to detect net-

work attacks and to limit any resulting damage, it is not sufficient to protect the network devices from the attacks. Therefore, it is also necessary to build a Network Intrusion Detection System (NIDS) in the network servers to detect the malicious network traffic efficiently.

Generally, the common NIDSs can be categorized into two types: Signature-based NIDS and Anomaly-based NIDS. Signature-based NIDS utilizes the pre-defined attack signatures corresponding to a known attack to classify the network traffic as a cyber attack. Thus, signature-based detection can be applied to detect known network traffic intrusion efficiently. However, it fails to detect some novel or zero-day attacks. In addition, its signature databases need to be updated frequently, so that it can maintain the good detection performance. As opposed to the Signature-based intrusion detection systems, Anomaly-based NIDS does not adopt the pre-defined signatures. Instead, the Anomaly-based NIDS utilizes anomaly detection techniques to detect the malicious network activities. As anomaly detection techniques aim at finding the unexpected pattern in the data, Anomaly-based NIDSs have the ability to detect both known and novel network attacks.

1.2.2 Principal Component Analysis based NIDS

Lakhina et al. propose a piece of pioneering work which introduced principal component analysis (PCA) based NIDS to detect network anomaly [68]. In this NIDS, PCA is adopted here to separate the network data into normal data and abnormal data. The square errors of the abnormal data will be utilized as the anomaly degrees to determine the anomalous data. Therefore, PCA based Network Intrusion Detection System has received lots of attention in recent years [1] [78] [57] [79] [17]. Some researchers attempt to improve detection accuracy by combining PCA with non-linear transform, for ex-

ample kernel [1] and Kullback-Leibler (KL) divergence [17]. In addition, [78] combines sketch data structure and PCA to identify the anomalous flows.

Unfortunately, most of these works detect the network anomalies only based on the traffic volumes. However, lots of network attacks nowadays are more subtle and cannot be detected by using these volume-based methods. Therefore, feature-based anomaly detection overcomes this limitation by generating a set of network traffic features, which can be derived by analyzing the packet header. In addition, PCA based network anomaly detection suffers from normal subspace limitation, which may trigger more false alarms. Moreover, PCA based network anomaly detection can only identify the anomalous time bins, but fails to provide the anomalous IP addresses.

To address the normal subspace contamination issue, the author proposes a Multi-Scale PCA (MSPCA) based detection algorithm which combined the wavelet analysis with PCA in [24]. The key intuition behind this method is to separate the normal and anomalous data at any time and frequency scale with the help of wavelet analysis. Therefore MSPCA ought to diminish the impact of normal subspace contamination. However in this Conventional-MSPCA algorithm, it is necessary to perform PCA at each scale of wavelet coefficients, which results in too many parameters and high time complexity. Hence, it is difficult to apply the Conventional-MSPCA algorithm to different network scenarios.

Based on the analysis of this Conventional-MSPCA algorithm, the author observes that there is no need to perform PCA at all the wavelet coefficient scales. The PCA-filter level can be controlled by the threshold of Multiscale Multivariate Energy Contribution Efficiency (MMECE). In addition, the author applies the Bayesian-PCA instead of the conventional-PCA. Compared with the conventional-PCA, the Bayesian-PCA has the advantages of less time complexity and the ability of automatic principal components (PCs) selection. Therefore, by introducing this MMECE threshold and

Bayesian-PCA, the author proposes an Improved-MSPCA algorithm with less parameter setting and lower time complexity. The author also proposes a novel feature-based detection system which incorporates the Improved-MSPCA with sketch data structure. Sketch data structure can help the proposed Improved-MSPCA algorithm identify the anomalous IP addresses, which solves the second limitation of PCA-based detection system.

1.3 Low Rate Denial of Service Attack

Most of the network attacks mentioned in Section 1.2.1 target at the network servers. Therefore, NIDS is built on the network server side to detect any possible attacks. However, there is one special class of DoS attack which sends periodic packets to attack the bottleneck routers. This attack is called as Low-rate Denial of Service (LDoS) attack. Thus, the detection of this LDoS attack is beyond the reach of NIDS of the network servers. Therefore, it is necessary to develop an efficient LDoS detection and mitigation algorithm in the bottleneck router side.

LDoS attacks mainly exploit TCP's congestion control mechanisms to degrade normal TCP senders' throughput. As there are two main types of congestion control mechanisms, LDoS attacks can be categorized into two kinds: Timeout-based LDoS [64] and Addictive-Increase Multiplicative-Decrease (AIMD) based LDoS [82]. Both LDoS attacks send periodically pulsing packets to the bottleneck routers to overflow their queues. Once their queues are full, these routers will drop the normal TCP packets. Therefore, the congestion window (CWND) size of TCP senders will be decreased according to the congestion control. For the Timeout-based LDoS attacks, these dropped packets will force the CWND size to be reduced to one packet. The victim TCP senders will experience the timeout state in every period of the LDoS attacks. The AIMD-based

Figure 1.2: Low Rate Denial of Service attack targeting at the bottleneck routers

LDoS attack will make the TCP senders halve their congestion window, so that they may enter the fast-recovery state. No matter what kind of LDoS attacks happens, the overall throughput of the normal TCP senders is quite limited, which seriously affects the user experience of network users. As the overall sending rate of LDoS attack is quite small, it poses a new challenge to network security.

Previous works attempt to detect the LDoS attack by exploiting its periodicity property. Some researchers propose time domain based LDoS detection methods, such as Dynamic Time Warping (DTW) [104], Multiple Sampling Averaging [128], Kalman filter [113], and flow connection entropy (FCE) decomposition [76]. Some other researchers explored frequency domain based detection methods to decompose the network signals so as to analyze their frequency domain properties. Several Fourier analysis based detection methods have been proposed [15, 23, 114]. Researchers also adopt wavelet analysis based LDoS detection [82, 115]. Furthermore, researchers utilize statistical based methods to detect LDoS attacks. [9] tests the detection performances

of several information entropies. In [8] and [2], Rank Correlation measures are utilized to discriminate the attack traffic.

Although many detection mechanisms have been proposed, their detection performances do not meet the expectations. In addition, most of these works just present the detection methods, but fail to provide the mitigation mechanisms. To address the LDoS attacks, the author first purposes one novel detection metric: Power Spectrum Entropy (PSE) which combines statistical analysis with signal processing techniques. The author then incorporates PSE into the Robust RED algorithm to design an efficient LDoS mitigation algorithm, named PSE based Robust RED (PRRED) queuing algorithm. This algorithm can be embedded into the bottleneck routers to detect and mitigate LDoS attacks efficiently and swiftly.

1.4 Image Outlier Detection

Figure 1.3: Example of image outlier detection. The images consist of normal images ('Cat') and other outlier images

Anomaly detection also plays an important role in computer vision. Here, anomaly detection is often referred to as outlier detection. Computer vision is a rapidly growing field that powers a variety of emerging techniques. Recently, deep learning methods have achieved great success in computer vision. As deep learning requires large training datasets, it has become more and more critical to build large image datasets for many computer vision tasks. Retrieving images from the Internet is one of the most useful methods [28, 58, 126]. However, the retrieved results often contain many irrelevant images. Figure 1.3 presents an example of image outlier detection. When we input 'Cat' query in a search engine, it may return a few dog images, which introduce outliers to the 'Cat' category. Therefore, outlier detection and removal are essential to help us construct pure training dataset.

Traditional anomaly/outlier detection techniques can also be applied to the image outlier detection. However, due to the high dimension of the image data, the traditional methods do not achieve good detection performance. Recently, due to the success of deep learning, many researchers adopt deep learning methods to detect image outliers [20, 40, 97, 99]. There exists a category of methods that utilizes the reconstruction error of autoencoders to detect outliers [4, 95, 116]. They are all based on the assumption that there are strong statistical correlations among the features of normal data. By minimizing the reconstruction errors of normal training data, autoencoder is able to capture these correlations. Therefore, normal data should have relatively smaller reconstruction errors than the outliers. Those data having large reconstruction errors are determined as outliers. However, existing works just train one autoencoder to detect outliers. For high-dimensional image data, it is very difficult for one single autoencoder to fully learn the feature correlations of normal data.

To address this problem, the author combines autoencoder with Adaboost (ADAE) to detect image outliers. By repeatedly training on the weighted normal dataset, ADAE

is able to obtain a sequence of weak autoencoders. The final prediction model of ADAE is a weighted summation of all the weak autoencoders. As each successive autoencoder attempts to capture the correlations that the previous one has failed to capture, ADAE can better capture the image feature correlations than the single autoencoder. In addition, the author introduces the Sparse Group Lasso (SGL) constraint to the learning objective of the autoencoder in order to eliminate unnecessary parameters. Furthermore, the author utilizes the Evolutionary Multi-Objective (EMO) based Optimization framework to determine the optimal penalty factors of SGL.

1.5 Contributions

The author has made the following contributions to address network anomaly detection and image outlier detection problems.

1.5.1 Detection of Network Anomalies using Improved-MSPCA with Sketches

The author first studies how to reduce the time complexity of MSPCA anomaly detection algorithm. By analyzing the energy distributions of the original network data matrix, the author finds that most of the energy is located in the wavelet coefficients of low frequency scales. Therefore, the author proposes the Multiscale Multivariate Energy Contribution Efficiency (MMECE) to control the PCA-filter scales. Furthermore, Bayesian-PCA is utilized to replace the conventional PCA to filter the wavelet coefficient matrices of each scale. Consequently, by introducing MMECE threshold and Bayesian-PCA, the proposed Improved-MSPCA can separate network data with less parameter setting and lower time complexity.

The author next proposes a novel network anomaly detection system based on the combination of the Improved-MSPCA algorithm and sketch data structure. This Sketch-MSPCA detection system is able to detect anomalous IP addresses. As the proposed system attempts to detect network anomalies based on flow-level features, it can reveal more network attacks than previous packet based detection system.

The author evaluates the Improved-MSPCA algorithm on DARPA99 dataset and tests the detection performance of the proposed Sketch-MSPCA detection system on MAWI dataset. The results show that the Improved-MSPCA algorithm overcomes the limitation of PCA based detection algorithm so as to achieve better anomaly detection performance. The proposed Sketch-MSPCA system is superior to the other related detection systems in terms of higher true positive rate and lower false positive rate.

1.5.2 Detection and Mitigation of LDoS Attacks using Power Spectrum Entropy

The author first introduces a novel LDoS detection metric: Power Spectrum Entropy (PSE) which combines signal processing technique with information entropy. As LDoS attacks can be modeled as periodic square signals, most of their energy is concentrated in the low-frequency range, whereas the energy of the normal flow signals is spread across a wider frequency. These two types of signals exhibit very different Power Spectrum distributions in the frequency domain. By incorporating Information Entropy with the Spectrum Analysis, it is easy to capture the difference between these two spectral distributions. Based on the adopted signal decomposition method, the author proposes two kinds of *PSE*: Fourier-PSE (*FPSE*) and Wavelet-PSE (*WPSE*). As their names imply, *FPSE* utilizes Fourier Transform to perform Power Spectrum Analysis, while *WPSE* takes advantage of Wavelet Transform.

Next, the author designs a PSE-based Robst RED (PRRED) queuing algorithm to detect and mitigate LDoS attacks, which involves two phases. In the first phase, the algorithm identifies a suspicious subset of packets based on their arrival time. In the second phase, these packets will be filtered based on the calculation of PSE. Therefore, this PRRED algorithm can help the bottleneck routers detect and mitigate the LDoS attacks efficiently and swiftly.

The author evaluates the PRRED algorithm mainly based on network simulations. The results reveal that PRED algorithm has a better LDoS mitigation performance compared to the traditional queuing algorithms, so that it can protect the throughput of the normal TCP senders.

1.5.3 Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection

The author proposes a novel image outlier detection algorithm which combines autoencoder with Adaboost (ADAE). One simple autoencoder is not enough to fully capture the feature correlations of normal data. Adaboost algorithm is adopted here to ensemble many autoencoders so as to enhance the detection performance. Each successive autoencoder is meant to compensate the reconstruction loss caused by the previous autoencoders. Therefore, ADAE can better learn the hidden structures of normal images so as to have a better outlier detection performance.

The author next introduces the Sparse Group Lasso (SGL) regularization term into the loss function of each autoencoder. Optimizing the loss function with SGL can make the whole detection model compact. In order to optimize this new loss function, the author combines Adagrad with Proximal Gradient Descent (Ada-PGD). Furthermore, the author utilizes the Evolutionary Multi-Objective (EMO) based Optimization

framework to obtain the optimal penalty factors of SGL.

By evaluating on several well-known image datasets, ADAE has been shown to have better detection performance than the other outlier detection methods. In addition, SGL regularization term helps to decrease the parameter number of the detection model.

1.6 Thesis Organization

The rest of the thesis is organized as follows:

- **Chapter 2** reviews the previous works on the related topics. Various network anomaly detection mechanisms are discussed. The detection schemes for LDoS attacks are presented. Furthermore, research about deep learning based image outlier detection is also studied.
- **Chapter 3** first improves the Conventional-MSPCA anomaly detection algorithm by introducing the MMECE threshold and Bayesian PCA. A sketch based MSPCA anomaly detection system is also described and the detection performance is analyzed and evaluated using DARPA and MAWI datasets.
- **Chapter 4** presents the LDoS detection and mitigation algorithm based on power spectrum entropy. The evaluation is conducted via network simulation.
- **Chapter 5** performs image outlier detection by combining autoencoder with Adaboost. SGL regularization term and EMO based optimization framework are introduced to make the detection model compact.
- **Chapter 6** concludes the thesis and sketches out future research directions.

Chapter 2

Literature Review

In this chapter, the author discusses the previous work related to network anomaly detection, LDoS attack detection and image outlier detection. Firstly, the literature about network anomaly detection is introduced. Secondly, the author reviews the literature on related advances in the detection and mitigation of Low-Rate DoS attacks. Finally, the author introduces the research on image outlier detection.

2.1 Network Anomaly Detection

Network anomaly detection techniques can be broadly classified into three main categories: statistical-based detection, machine learning-based detection and signal processing-based detection [46, 69].

2.1.1 Signal Processing based Anomaly Detection

From the perspective of signal processing, network anomalies are treated as abrupt changes in the time series data. Therefore, many signal processing techniques have been

proposed here to detect the anomalies in the network traffic data. In this subsection, the author lists several well-known signal processing based network anomaly detection methods.

In [7], Barford et al. propose a Discrete Wavelet Transform (DWT) based detection method. In the first step, they aggregate SNMP and IP flow data into time series signals. DWT is utilized to decompose the network time series into three parts: L-part, M-part, and H-part. Then based on the deviation scores of these three parts, the anomaly could be detected. To detect the short-lived anomalies, the detection system only analyzes the H-part. For the long-lived anomalies, the L-part and M-part need to be considered. The evaluation results show that this DWT-based detection method could isolate both short-lived and long-lived anomalies efficiently.

In [31], Dainotti et al. propose an automated system to detect DoS attack. The whole detection system has a two-stage architecture which combines Cumulative Sum (CUSUM) with Continuous Wavelet Transform (CWT). The statistical anomaly detection techniques are utilized in the first stage to perform the rough detection. An adaptive threshold algorithm and CUSUM algorithm are proposed in this stage. Once an alarm is raised in this stage, it will trigger the second stage: Fine Detection module. Based on the CWT coefficients of signals, the anomaly can be determined. This two-stage detection system is able to detect the network anomalies accurately and reduce the false alarms.

In [54], Huang et al. propose a real-time network anomaly detection framework, called Waveman. They utilize the percentage deviation and entropy of the packet number as the detection features. By evaluating the detection system on DARPA and EnetRegistry datasets, they compare the detection performance of four different families of wavelets. The evaluation results show that Coiflet and Paul wavelets have the best detection performance.

In [80], Lu et al. combine the wavelet analysis with AutoRegressive eXogenous (ARX) model to detect network anomalies. Several flow level features are explored here. For each feature, its corresponding wavelet coefficients are fed into the ARX model for training. Based on the least square errors, the optimal parameters of the ARX model can be estimated. In the test phase, any data with high residual error will be determined as the anomaly. This detection model has the best detection performance when compared to unsupervised clustering detection and nonparametric CUSUM based SYN flooding detection.

In [18], Callegari et al. also combine CUSUM with wavelet analysis to detect the anomalies. The steps of this detection system are the reverse of the detection steps in [31]. As the performance of CUSUM is affected by the seasonal trends of network traffic, they utilize wavelet transform to filter the seasonal trend. CUSUM is utilized in the next step to perform the final detection. This detection system is tested on a public dataset collected in the Abilene/Internet2 Network. The results reveal that the wavelet preprocessing step does improve the detection performance of CUSUM algorithm.

In [87], Palmieri and Fiore adopt the recurrence quantification analysis (RQA) to detect network anomalies. RQA is firstly introduced by Zbilut and Webber in [122]. It is often utilized to analyze the non-stationary properties and hidden time correlations of statistical time series. Palmieri and Fiore input the ROA outputs to the SVM to classify the abnormal data. This anomaly detection method has been proven to have a good detection performance in [87].

2.1.2 Statistical based Anomaly Detection

Researchers have also proposed many anomaly detection systems based on statistical analysis.

For some of the statistical-based methods, they attempt to fit a pre-defined distribution on the training data. Thus any data with low probability under this distribution will be determined as an outlier. In [41], Eskin develops a probability anomaly detection method. In [38], Dukkupati et al. apply a mixture model based anomaly detection using q -Gaussian distributions. The parameters of this model are estimated by Expectation Maximization based Maximum Likelihood Estimator. The detection performances of these methods highly depend on the choice of distributions. An unsuitable distribution may result in a bad detection performance.

Brutlab [14] proposes a network anomaly detection method based on Holt Winters forecasting algorithm. In this paper, the byte rate is utilized as the statistical feature of network traffic. Based on the previous byte rates, Holt Winters algorithm can predict the next byte rate value in the time series. The anomalies are identified if the current byte rate differs too much from the predicted value. The main drawback of this method is that it just utilize the byte rate as the detection feature. As network attacks nowadays become more diverse, it is hard to identify all the attacks by monitoring only the byte rates.

An unsupervised Kalman filter-based anomaly detection is proposed by Soule et al. [103]. At the beginning, all the traffic flows are aggregated into Origin-Destination (OD) flows. Then these flows are stored into a matrix. Based on the OD matrix of previous time bins, the method can estimate a predicted OD matrix in the current time bin by using Kalman filter. Therefore, if the predicted OD matrix deviates too far from the real OD matrix, the detection system can determine the network anomalies. However, this method can only identify the occurrence of network anomalies, but fail to provide the details of the anomalous flows.

Nychis et al. [86] adopt Shannon entropy to detect network anomalies. In this method, the extracted statistical features are the entropies of source IP, destination IP,

source port, destination port, flow size, and host degree distributions. The anomalies are detected in the entropies using a heuristic approach.

[48] proposes a Maximum Entropy based network anomaly detection technique. At the beginning, the detection method learns a density model by analyzing the baseline traffic distribution using Maximum Entropy. In the test phase, the detection system computes the relative entropy of the incoming traffic with respect to the baseline traffic. Therefore, this technique can identify the anomalies based on these relative entropies.

2.1.3 Machine Learning based Anomaly Detection

In machine learning, anomaly detection is referred to the detection of data which do not conform to an expected pattern or to other data in the dataset. Many well-known machine learning algorithms have been applied in network anomaly detection recently. Generally, machine learning based anomaly detection can be classified into two main categories: Supervised learning based and Unsupervised learning based anomaly detection.

Supervised Learning based Detection

Supervised learning methods require a labeled training dataset. Therefore, the supervised detection methods have better detection performance than the unsupervised detection methods.

[72] presents a novel network anomaly detection method based on improved Transductive Confidence Machine for K-Nearest Neighbors (TCM-KNN). The evaluation on the KDD99 dataset shows that TCM-KNN can effectively detect network anomalies with high accuracy and few false alarms. In addition, the proposed method is robust when faced with unclean data.

[85] cascades the K-means clustering and the C4.5 decision tree algorithm to detect network anomalies. K-means clustering algorithm is utilized here to cluster the training data into k clusters. In each cluster, C4.5 decision tree algorithm is applied to learn a density region of normal data. Finally, the final decision is made by concluding the results from all clusters.

In [84], Mylay et al. propose a Decision Tree based multiclass Support Vector Machines (SVM) algorithm to detect network anomalies. They train five SVM models for the five categories of data. Decision Tree is applied next to decide which label this datum belongs to. The evaluation results on the KDD99 dataset show the superiority of this detection method.

[53] presents a robust support vector machine (RSVM) based network anomaly detection. RSVM addresses the over-fitting problem by making its decision surface smoother. In addition, the amount of regularization terms of RSVM can be controlled automatically. In RSVM, the number of support vectors is less than that of the conventional SVM. Therefore, the running time of RSVM is much faster compared to the conventional SVM. The evaluation experiments are conducted on the 1998 DARPA BSM data. The detection performance of RSVM is much better than that of conventional SVM and nearest neighbor.

In [56], Niyaz et al. propose a deep learning based network intrusion detection system. They utilize the Self-taught Learning (STL), which is a deep learning technique. STL consists of two stages. In the first stage, by adopting the unsupervised feature learning method, a pre-trained classification network can be obtained. In the second stage, the pre-trained network is applied to the labeled training data. Therefore, one can get the final classification network. Its detection performance is comparatively better than that of other anomaly detection methods.

Researchers have also proposed several neighbor-based detection methods. Neighbor-based methods assume that normal data has relatively more neighbors than the outlier data [13, 63]. In [13], Breunig et al. adopted a density-based local outlier factor (LOF) to address this issue. This LOF only considers a restricted neighborhood of each data. Therefore, LOF can identify the local outliers. In addition, by normalizing LOF, Kriegel et al. [63] propose the local outlier probabilities to detect outliers. However, the search for the nearest neighbors prohibits such methods to be applied to high-dimensional data due to the curse of dimensionality. High dimensional data will fool the algorithm to locate the improper neighbors, which will decrease the detection accuracy.

In [75], Liu et al. propose another tree-based method that can explicitly isolate anomalies. They adopt the random forest technique to perform the classification. Anomalies can be isolated close to the root of the tree, whereas the model isolates the normal data at the end of each tree. Based on this property, the anomalies can be easily identified.

One Class Classification based Detection

One class classification is a special type of supervised learning. It learns the model from a training dataset which only consists of data from one class. In the real situation, it is much easier for us to obtain the normal data rather than the abnormal data. Therefore, one can build a training dataset full of normal data to train a one class classification model to detect anomalies.

One class SVM (OCSVM) is one of the most well-known one class classification algorithm [98]. In [98], Schölkopf et al. attempt to estimate a subset S which contains all the normal data. They try to estimate a function f which is positive on S and

negative on the complement. This problem can be transformed into the Quadratic Programming problem, which is similar to the original SVM problem. Therefore, they propose the one class support vector machine (OCSVM). In [39], Azami et al. combine the support vector data description (SVDD) model with the calibration functions. Thus, the output of SVDD can be transferred into the outlier probabilities. Quinn et al. [91] attempt to find the boundary of normal data based on a squared-loss function which is similar to OCSVM method.

All of these three methods [39, 91, 98] can utilize kernels to obtain the nonlinear detection methods. However, the kernel calculation is quite time consuming. Thus, these methods are both time-consuming in training and testing.

2.1.4 Unsupervised Learning based Detection

Contrary to supervised learning, unsupervised learning methods do not require the labeled training dataset. The unsupervised learning based anomaly detection methods are normally based on two basic assumptions. Firstly, most of the network data are from normal traffic, whereas the abnormal data only constitutes a small portion. Secondly, the abnormal data is statistically different from the normal data.

In [92], Dennis Irwin proposes a Self-Organizing Maps (SOM) based network anomaly detection. SOM can be defined as a manifold-based dimension reduction method. Thus, SOM can compress statistical information from a high dimensional space into a low dimensional space. As most of the network data are from normal traffic, SOM model can capture the statistical information of the normal traffic. Therefore, those data with high SOM errors are the abnormal data.

In [77], Liu et al. attempt to optimize a kernel-based max-margin objective to address the image outlier detection problem. For inliers and outliers, a soft label

assignment will be learned by this optimization. The algorithm is evaluated on four well-known datasets. The results show the detection efficiency and robustness of the proposed algorithm.

Lakhina et al. [68] aggregate all the network traffic into Origin-Destination (OD) flows. For each flow, the traffic volume is generated every 5 minutes to get the traffic matrix. PCA-based detection method is then proposed to separate the normal traffic from the original traffic matrix. Then, the square error between the original and normal traffic matrices are utilized to determine the anomalous data. This PCA-subspace detection method has received quite a lot attention all these years [1] [78] [57] [93] [12]. [50] highlights the advantage of PCA as an unsupervised detection method. In [79], a distributed PCA is applied to detect anomalies in wireless sensor network. Callegari et al. combine PCA with Kullback-Leibler divergence to improve the detection accuracy [17]. Although PCA has been widely applied in the field of network anomaly detection, Ringberg et al. address two critical limitations of PCA-based method in [93].

The first limitation is normal subspace contamination, which could trigger more false alarms. To address this limitation, many researchers proposed several extensions to the PCA-based detection system. In [12], Brauckhoff et al. applied the Karhunen-Loeve expansion to capture temporal correlations. In [30, 88, 94], the authors adopted the robust PCA-based detector so that the system could reduce the impact of outliers. While in [19], Camacho et al. attempt to combine Multivariate Statistical Process Control (MSPC) theory with PCA to overcome this limitation. Nevertheless, almost all the extensions of PCA-based anomaly detection try to perform PCA in the original time scale, which cannot separate anomalous data efficiently. The author has proposed a MSPCA-based detection algorithm by utilizing PCA to filter the data matrix at any time and frequency scale [24] [25]. However, this MSPCA-based algorithm has so many parameters that it is difficult to obtain the optimal parameter setting. Meanwhile, the

time complexity of this algorithm is very high.

The second limitation of the conventional-PCA method is that PCA-based detection algorithm can only point out the anomalous time bins but fails to tell which IP addresses are responsible for these anomalies. The works in [60,61,71] give a remarkable approach which combines PCA-based detection algorithm and sketch data structure. Therefore, their systems are able to identify the anomalous IP addresses. However, these systems adopt PCA-based detection algorithm, hence they still face the problem of normal subspace contamination, which results in high false positive rate. Moreover, their systems are packet-based detection system. As network attacks nowadays become more diverse, it is hard to identify all the attacks by monitoring only the packet count.

In Chapter 3, the author first proposes an Improved-MSPCA anomaly detection algorithm which applies Bayesian-PCA to filter the data of each scale instead of Conventional-PCA. Thus the algorithm could not only reduce the time complexity, but also determine the effective principal component number automatically. In addition, the Improved-MSPCA applies MMECE threshold to limit the PCA-filter level which is able to reduce the time complexity. The author also proposes a new feature-based detection system which combines sketch data structure with the Improved-MSPCA detection algorithm. In this novel detection system, several flow-based features are aggregated to characterize the network events. Meanwhile, the combination of sketches and Improved-MSPCA enables the proposed system to identify the anomalous IP addresses in addition to the detection of a possible anomaly at a particular time instant.

2.2 Detection of Low-Rate Denial of Service Attacks

Low-Rate DoS (LDoS) attacks were first reported by Kuzmanovic in [64]. In their work, they mainly discuss one type of LDoS attacks which exploits TCP's retransmission

time-out mechanism to reduce Quality of Service (QoS) of TCP flows. [82] summarizes another kind of LDoS attacks which targets TCP's Additive-Increase-Multiplicative-Decrease (AIMD) mechanism. Both of these two attacks send periodic packet bursts to the bottleneck router to flood them so as to drop normal TCP packets periodically. These periodic packet drops can result in the reduction of congestion window sizes of normal TCP senders. Therefore, the throughput of the normal TCP senders is quite limited. As the overall sending rate of LDoS attacks is quite small, it poses a new challenge to network security.

Many researchers have also conducted studies to analyze LDoS attacks. In [49], Guirguis et al. develop control-theoretic models to analyze LDoS attack. Yajuan Tang et al. analyze LDoS attack from the aspect of feedback control [108]. In [81], the authors propose a mathematical model to estimate attack effects of LDoS.

In recent years, researchers have also proposed many LDoS detection methods. Existing LDoS detection methods can be classified into three main categories: Time Domain based Detection, Frequency Domain based Detection, and Statistical based Detection .

2.2.1 Time Domain based Detection Methods

LDOS attackers send periodic pulse packets to the bottleneck routers, which results in strong periodicity in the time domain. Therefore, researchers propose several time domain based mechanism to detect this periodicity.

In [104], Sun et al. present a time domain based detection method to determine the periodical pattern of LDoS attacks. In this work, the traffic is sampled at a constant rate to generate time series signals at the bottleneck routers. The periodic features of an input signal are extracted by utilizing auto-correlation. In order to compare

the auto-correlations of LDoS attack signal with that of normal TCP signals, Sun proposes the Dynamic Time Warping (DTW). The lower the DTW value, the higher the similarity degree of the input signal as compared to the LDoS signature. It is to be noted here that this LDoS signature needs to be predefined. Therefore, based on the value of DTW , the LDoS attack flow can be identified. When the existence of a LDoS attack is determined, Sun et al. propose a deficit round robin (DRR) algorithm to mitigate the LDoS attacks.

There are two main drawbacks of DTW detection method. Firstly, the time complexity of the DTW detection is quite large. Assuming the length of time signal is n , the time complexities of auto-correlation and DTW are both $O(n^2)$, which is quite time-consuming. Secondly, DTW compares each incoming flows with a predefined LDoS signature. Once the LDoS attackers change the attacking parameters, DTW will not capture these LDoS variations. Due to this drawback, the DRR mitigation algorithm can only protect 80% of the normal TCP throughput.

[128] proposes a novel LDoS detection method by using Multiple Sampling Averaging Based on Missing Sampling (MSABMS). This is the first time the model of small signal is applied to LDoS detection. Furthermore, they also establish an eigenvalue-estimating matrix to estimate the attack period.

In [124], Zhang et al. propose a novel LDoS detection metric - Congestion Participation Rate (CPR). CPR can calculate the percentage of network flows involved in the network congestion. Normal TCP flows can avoid the network congestion because of the congestion control mechanism. Therefore, the CPR of normal TCP flow normally has a small value. As LDoS can generate congestion to limit the network throughput, its CPR value will be relatively larger. Based on CPR values, one can easily identify the LDoS flows. Furthermore, [124] also proposes a CPR-based LDoS filtering system which can be embedded into the bottleneck routers.

2.2.2 Frequency Domain based Detection Methods

The strong periodicity of the LDoS attacks means that in the frequency domain, the LDoS signals will appear different compared to the normal TCP signals. Based on this property, researchers have developed several frequency domain based LDoS detection mechanisms.

Fourier Analysis based Detection

By adopting the fourier transform, the traffic signals can be decomposed to the frequency domain. Thus, the power spectrum density (PSD) can be calculated. By comparing the PSD of LDoS attack signal with that of normal TCP flow, [23], [114] and [15] find out that the energy distributions of LDoS and normal TCP flows are quite different. For LDoS attack, its energy is mostly located in the low frequency domain whereas the normal TCP flow has a very flat energy distribution over a wider frequency domain.

[23] computes the *cumulative traffic spectrum* (CTS). For each incoming flow, CTS is calculated by summing up the PSD values in the frequency range $[0, p]$, where p is the pivot frequency. As most of LDoS energy is located in the low frequency domain, the CTS values of LDoS signals are comparatively larger than the CTS values of normal TCP signals. Therefore, based on the CTS values, the detection system can identify which flow is the LDoS attack flow. The detection performance of the CTS based detection highly depends on the choice of the pivot frequency. Before the detection, this pivot frequency should be selected by finding the maximum gap point between the CTSs of attack flows and normal TCP flows. However, the changes in LDoS attack duration and period may affect the maximum gap point, which means that different LDoS attacks may have different pivot frequencies. Therefore, this preset

pivot frequency cannot guarantee the detection performance when it encounters LDoS with different attack duration and period.

In [114], Wu et al. attempt to utilize the infinite impulse response (IIR) filter to mitigate the LDoS attack flows. This detection method is also based on the fact that most of the LDoS energy is located in the low frequency range. The IIR filter performs as a high-pass filter here. Therefore, the low frequency signals will be filtered out by IIR filter. In the meantime, most of the normal TCP signals will be preserved. However, similar to the work in [23], this mitigation method is not robust to different kinds of LDoS attacks. When the attacker changes the attack duration or period, some of the attack packets can still pass through the IIR filter. The throughput of the normal TCP senders is still limited.

Wavelet Analysis based Detection

Several researchers also propose LDoS detection methods based on wavelet analysis. In [82], Luo et al. introduce a method to detect the LDoS attack which combines wavelet analysis and Cumulative Sum (CUSUM) to capture the variability in network traffic. In the first step, this detection system adopts discrete wavelet transform (DWT) to analysis incoming network traffic. DWT is very good at detecting the variation of the network traffic distribution. The second step is based on CUSUM. By analyzing the wavelet features extracted by DWT, CUSUM will determine the LDoS attack flows. This method does provide very high detection accuracy, but its detection latency is too high (30s). Thus it cannot be applied to real-time detection.

In [115], Wu et al. utilize wavelet analysis to calculate the *Hölder* exponents, which reflects the self-similarity of the network traffic. Research result in [43] shows that aggregated network traffic has self-similarity, whereas LDoS attack flow can disrupt

this property. Therefore, the *Hölder* exponent value of network traffic will be changed when there are LDoS attacks in the network traffic. Therefore, LDoS attack can be detected based on the change in *Hölder* exponents. However, there is a drawback in this method. The multifractal property appears in the aggregated network traffic. For the single network flow, it does not appear to have strong multifractal property, which may result in false detection. Therefore, this method does not give very promising result. Its false positive rate is too high. In addition, the paper just provides the detection method, but not the LDoS mitigation method.

2.2.3 Statistical based Detection

Some statistical analysis techniques have also been adopted to detect LDoS attacks. In [117], Xiang et al. propose a low-rate DoS detection method based on the generalized entropy metric and the information distance metric. They sample the network traffic and calculate its corresponding probability distribution in each router. Then, the information distance between these distributions can be calculated. The LDoS attacks could be determined based on the distance. The evaluation results in [117] show that the proposed metric clearly reduces the false positive rate compared to the baseline false positive rate. However, this paper fails to provide the baseline false positive rate so that one does not know the real value of the reduced false positive rate. In addition, [117] does not propose any LDoS mitigation method.

Bhuyan et al. [9] exploit several kinds of information entropies to detect low-rate and high-rate DoS attacks. They conduct the experiments in MIT Lincoln Laboratory, CAIDA DDoS 2007 and TUIDS DDoS datasets. In this paper, they just present the comparisons of entropies of normal TCP flows and LDoS flows, but fail to provide the detection true positive rate and false positive rate. In addition, Bhuyan et al. do not

propose a LDoS mitigation method.

In [2], Ain et al. utilize rank correlation measures to detect LDoS attacks. In their work, they adopt two rank correlation measures: Spearman Rank Correlation (SRC) and Partial Rank Correlation (PRC). The evaluation results reveal that PRC has a better detection performance than SRC.

2.2.4 Other Detection Methods

In addition to the previously mentioned detection methods, there are also some other LDoS detection methods which are published in recent years. [64] explores the use of RED-PD and *min-RTO* randomization to mitigate LDoS attacks. However, both methods are not sufficient to fully mitigate the attacks. Shevtekar et al. [101] propose a light-weight mechanism to capture the periodicity of the traffic flows. However, as the short-lived TCP flows also exhibit a periodicity pattern as well, the mechanism may result in many false alarms. In [67], Kwok et al. introduce a detection system called Halting Anomaly with Weighted choKing (HAWK) in which they record the flow statistics into a flow table. This method can protect the normal traffic from the single LDoS attacks, but it is not able to identify the distributed LDoS attacks.

In conclusion, many researchers have proposed LDoS detection system by utilizing different kinds of techniques. However, for most of the LDoS detection systems mentioned above, they only present the detection methods but fail to provide the LDoS mitigation mechanism. In [125], Zhang et al. propose a Robust Random Early Detection Queuing algorithm which can be applied to the routers to mitigate LDoS attacks. NS-2 simulation results revealed that this algorithm has a better mitigation performance compared to other traditional queuing algorithms. However, as this algorithm only has a very rough detection of LDoS attacks, it may raise many false alarms which

result in more normal TCP packet drops.

In Chapter 4, the author first introduces two novel information metrics to detect LDoS attacks: Fourier Power Spectrum Entropy (*FPSE*) and Wavelet Power Spectrum Entropy (*WPSE*). These two metrics utilize Fourier and Wavelet Analysis to perform Power Spectrum Analysis respectively. Then, by combining Power Spectrum Analysis with Information Entropy, it will become easy to distinguish the LDoS attack flows from the normal flows. As *PSE* utilizes all the Power Spectrum Values, these two detection metrics are more robust to detect the different LDoS attacks. In addition, by incorporating *PSE* with Robust RED algorithm [125], the author also proposes a novel LDoS attack filtering algorithm: Power Spectrum Entropy Robust RED (PRRED) algorithm. It combines the *PSE* detection metric with the Robust RED queuing algorithm. As *PSE* can determine LDoS attacks efficiently, PRRED is thus able to detect and mitigate the LDoS attack packets swiftly and efficiently.

2.3 Image Outlier Detection

In the field of image processing or computer vision, anomaly detection is also referred to as outlier detection. Most of the methods mentioned in Section 2.1 can also be applied to the image outlier detection. Recently, with the popularity of deep learning, many researchers apply the neural networks and deep learning techniques to image outlier detection.

In [20], Carrera et al. utilize the convolutional sparse models to learn a dictionary of filters, which can capture the structure of normal images. In the test phase, each test image will be passed into these filters. The indicator vectors can be calculated based on these encoded outputs. The image outliers can be determined by using these indicators.

[40] combines the deep belief networks (DBNs) with the one-class SVM. At the beginning, an unsupervised DBN is trained to extract the deep learning features of each image. OCSVM is trained based on these features. DBNs is a deep learning architecture, which consists of multiple layers. After training on the normal images, DBN is able to capture their structures so as to generate higher level features of the images. These features enable OCSVM to have a better detection performance. Similar to [40], in [99], Seeböck et al. utilize a Deep Convolutional Autoencoder (DCAE) to extract deep learning features and feed them into OCSVM.

Recently, Generative Adversarial Network (GAN) has been applied to image outlier detection [97]. GAN is introduced by Goodfellow in [47]. It can learn a generative model to generate realistic images. GAN normally consists of two neural networks: the generator and the discriminator. The ability of the generator network is to generate new image data, whereas the discriminator network is meant to classify the real images. By performing an alternative optimization, the generator can fully capture the probability distribution of the training data so that it can generate new images from this distribution. GAN-based outlier detection in [97] is a one-class classification model. At the beginning, GAN is trained on the normal datasets. In the test phase, the outlier image can be determined based on the residual score and the discrimination score.

In [95], Sakurada et al. propose an Autoencoder-based outlier detection method. As autoencoder can capture the nonlinear correlations as well as the linear correlations, it has better detection performance than PCA-based detection methods. However, when applying the autoencoder to image outlier detection, the detection performance is not always very palatable. This is because the single autoencoder fails to fully capture the correlations among features, especially in the high-dimensional datasets, resulting in poor detection accuracy.

In [116], Xia et al. propose another autoencoder-based image outlier detection algorithm. They observe that the reconstruction errors of a well-trained autoencoder can separate the inliers and outliers easily. Thus, in order to make the reconstruction errors more separable, a discriminative objective is introduced to the overall loss function. Therefore, the discriminative labeling can be integrated into the learning process. The outliers can be determined automatically during the optimization.

In Chapter 5, the author first addresses the image outlier detection problem by combining autoencoder with Adaboost. The overall idea of this algorithm is to sequentially apply the weak autoencoders to repeatedly modified versions of the training data. Therefore, by using an ensemble of these weak autoencoders, Adaboost-Autoencoder (ADAE) is able to better capture the statistical correlations among the features of the normal data so as to achieve better detection performance. In addition, by introducing Sparse Group Lasso constraint, the sparse solution can be generated for the sake of making the detection model simpler. The best penalty factors of SGL are determined by the evolutionary multi-objective optimization method.

Chapter 3

Detection of Network Anomalies using MSPCA with Sketches

This chapter presents MSPCA (Multi-Scale Principal Component) algorithm to detect computer network anomalies. MSPCA is an extension of PCA which merges PCA and wavelet analysis. The Conventional-MSPCA is first introduced to network anomaly detection in [24]. As MSPCA performs PCA in each scale of wavelet coefficient matrix, it can overcome the normal subspace contamination. However, in this Conventional-MSPCA algorithm, PCA is utilized to filter each scale of wavelet coefficient, which results in too many parameters and high time complexity. Therefore, it is difficult to apply this Conventional-MSPCA algorithm to different network scenarios.

Based on the analysis of Conventional-MSPCA algorithm, the author observes that there is no need to perform PCA at all the wavelet coefficient scales. The author suggests that the PCA-filter level can be controlled by the threshold of Multi-scale Multivariate Energy Contribution Efficiency (MMECE). In addition, the author applies Bayesian-PCA instead of conventional-PCA, as Bayesian-PCA is able to select the effective number of principal components automatically and is less time-consuming.

Therefore, by introducing this MMECE threshold and Bayesian-PCA, an Improved-MSPCA is proposed with less parameter setting and lower time complexity. The author also proposes a novel feature-based detection system which incorporates Improved-MSPCA with sketch data structure. This system helps to identify the anomalous IP addresses.

The author's contributions are as follows:

- An improved-MSPCA anomaly detection algorithm with less parameter setting and lower time complexity is proposed. The proposed algorithm has the ability to reduce the impact of normal subspace contamination.
- A novel feature-based Sketch-MSPCA detection system is proposed to identify the anomalous IP addresses. This detection system combines sketch data structure with MSPCA algorithm.
- The author compares the performance of this novel detection system with that of other related detection system by ROC curve criteria. By testing the detection system on the MAWI dataset, a great improvement is achieved compared to two other detection systems in [60] and [61].

The rest of this chapter is organized as follows. In Section 3.1, the author illustrates conventional Multi-Scale PCA based anomaly detection algorithm and its drawbacks. In Section 3.2, the author discusses in detail the Improved-MSPCA detection algorithm and its advantages. In Section 3.3, the author evaluates the Improved-MSPCA detection algorithm on the DARPA 1999 datasets and compares the results with other PCA-based algorithms. In Section 3.4, the author introduces the detailed mechanism of Sketch-MSPCA anomaly detection system. In Section 3.5, the author performs an extensive evaluation of the proposed detection system on the more recent MAWI datasets. Section 3.6 concludes this chapter.

3.1 Conventional-MSPCA

Principal Component Analysis (PCA) is a well known dimensionality-reduction technique. Lakhina et al. [68] use PCA to separate the network traffic data into normal subspace and residual subspace. They assume that the anomalous information all falls into the residual subspace. As a result, they determine an anomalous time point when its corresponding residual data exceeds some threshold. However, as shown in [93], the biggest challenge of PCA is the normal subspace contamination. After PCA computing, some large anomalies may still be located in the normal subspace. Therefore, it is difficult for PCA algorithm to obtain a clean normal subspace data, which will result in more false alarms [123].

Multi-Scale Principal Component Analysis (MSPCA) is an extension of PCA which merges PCA and wavelet transform. It is introduced by [6] to diagnose the system faults. Recently, it has been applied to network intrusion detection [24].

The intuition of the Conventional-MSPCA is to perform PCA to filter the wavelet coefficient matrix of each scale, which is good to clean up the normal subspace data. The main procedures of MSPCA are shown in Fig 3.1. MSPCA performs PCA on the coefficient matrices of different levels simultaneously. A PCA is also performed on the final reconstructed matrix.

Figure 3.1: Procedures of Conventional-MSPCA

MSPCA separates the original data matrix X into normal data matrix $\tilde{X}'$ and anomalous data matrix $\hat{X}$. In Fig. 3.1, $\tilde{X}$ and $\tilde{X}'$ can be calculated as: $\tilde{X} = \tilde{W}^T W X$, $\tilde{X}' = \tilde{X} V_k V_k^T$, in which $\tilde{W}^T$ is the simplified wavelet reconstruction matrix and V_k is the final PCA loading matrix with k selected PCs. Thus $\tilde{X}'$ is the normal subspace data and the anomalous subspace data can be calculated as $\hat{X} = X - \tilde{X}'$. By performing PCA to each scale's wavelet coefficient, this Conventional-MSPCA is expected to separate the normal and residual subspaces more efficiently. As a result, MSPCA can restrain the impact of normal subspace contamination.

However, there are two main drawbacks of this conventional-MSPCA when applied to network anomaly detection:

High Time Complexity In each wavelet scale of the decomposed network data, the conventional-MSPCA usually utilizes the conventional-PCA whose computational complexity is $O(nm^2)$ for the data matrix with the size $(n \times m)$. For the wavelet transform of a n -length signal, the time complexity is $O(n)$. As seen in Fig. 3.1, the author needs to perform wavelet transform m times to decompose the data matrix into level L and to operate PCA $(L+2)$ times. Thus, the total time complexity of MSPCA is $O(nm + (L+2)nm^2)$, which tends to $O(Lnm^2)$. Therefore for a large dataset, the conventional-MSPCA is quite time-consuming.

Difficulty in Tuning its Parameters The Conventional-MSPCA consists of wavelet analysis and multiple PCAs. Therefore, the author needs to select an appropriate mother wavelet, set wavelet decomposition level, and determine the number of PC retained in each wavelet decomposition scale. It is easy to set the two parameters of wavelet analysis: mother wavelet and decomposition level. The Symlet wavelets which have compact support abilities and increased symmetry are good for singularity detection [22] and are robust for MSPCA [119]. Therefore, the Symlet wavelets are commonly adopted for the conventional-MSPCA method. In [6], the author finds that

$L = \log_2 n - 5$ is a good choice for decomposition level. However, there are $(L+2)$ times of PCA in MSPCA, so the author needs to determine the number of retained PCs for each PCA, which results in many combinations of PC. It is difficult to select the best combination. In [118], the author applies a cross-validation method to select the best combination. However, it is quite time consuming to perform this cross-validation.

Due to these two major weaknesses, it is difficult to apply the Conventional-MSPCA directly to traffic anomaly detection. In the following section, the author will propose the improved-MSPCA algorithm to overcome these two weaknesses so that it can be adopted to the traffic anomaly detection.

3.2 Improved-MSPCA based Intrusion Detection Algorithm

In this section, the author will provide detail description of the Improved-MSPCA detection algorithm. Compared to the Conventional-MSPCA, there are two improvements. First, the PCA-filter scales are controlled by the threshold of Multi-scale Multivariate Energy Contribution Efficiency (MMECE). Second, Bayesian-PCA is adopted to filter the wavelet coefficient matrices. Consequently, the Improved-MSPCA can separate network data with less parameter setting and lower time complexity.

3.2.1 Multiscale Multivariate Energy Contribution Efficiency (MMECE)

In the Conventional-MSPCA, L level decomposition results in 1 approximation coefficient matrix $(H_L X)$ and L detail coefficient matrices $(G_j X, j = 1, \dots, L)$. The energy

Table 3.1: Energy Contribution of each level in traffic data

	H_6X	G_6X	G_5X	G_4X	G_3X	G_2X	G_1X
DARPA Week5 Day1	70.66%	16.39%	6.25%	2.39%	1.47%	1.34%	1.50%
MAWI 20130713	70.98%	13.13%	6.94%	3.20%	2.15%	2.02%	1.57%
MAWI 20140614	79.73%	7.04%	5.99%	3.72%	1.84%	1.03%	0.65%
MAWI 20150502	64.76%	17.49%	6.49%	6.13%	3.13%	1.17%	0.83%

contribution of each wavelet decomposition level can be determined based on the multiscale multivariate energy contribution efficiency (MMECE):

$$\begin{aligned}
 MMECE_{H_LX} &= \frac{tr[S_{H_LX}]}{tr[S_{H_LX}] + \sum tr[S_{G_jX}]} \\
 &= \frac{E_{G_jX}}{E_{H_LX} + \sum E_{G_jX}}
 \end{aligned} \tag{3.1}$$

$$\begin{aligned}
 MMECE_{G_jX} &= \frac{tr[S_{G_jX}]}{tr[S_{H_LX}] + \sum tr[S_{G_jX}]} \\
 &= \frac{E_{G_jX}}{E_{H_LX} + \sum E_{G_jX}}
 \end{aligned} \tag{3.2}$$

Where S_* represents the covariance of each matrix. Thus, the energy of each level is derived from the trace of covariance matrix. The author selects the network traffic data of four days as an example to present the energy contribution of each level, which is shown in Table 3.1.

It is seen that for network data, the wavelet coefficient matrices of low-frequency parts (H_6X, G_6X, G_5X) contribute the most amount of energies whereas the matrices of high-frequency parts (G_4X, G_3X, G_2X, G_1X) only occupy a very small percentage of total energy.

In addition to these four days of traffic, the network traffic traces of another 30 days are randomly selected from the MAWI dataset. The results of the energy contributions of these network traces show similar trend. For all of the selected data, the wavelet coefficient matrices of low-frequency parts (H_6X, G_6X, G_5X) occupy more than 80% of the total energy.

In the area of network anomaly detection, researchers often assume the abrupt changes in network traffic as anomalies. From the perspective of signal processing, the abrupt changes are interrelated with high-frequency signals. This is why a small number of PCs are selected for the high-frequency matrices to separate the anomalous information in the Conventional-MSPCA. However, as these high frequency matrices mostly contain information of anomalies, there is no need to perform PCA to them. All these coefficients can be set to zero, which saves the computational time and is equal to selecting 0 PC numbers for these matrices. Meanwhile, the algorithm will not lose much information of the normal subspace as the energy occupation of high-frequency part is very small. Therefore, setting the coefficients of the high-frequency parts to zero will not affect the generation of normal data, whereas it will reduce the time complexity of the entire algorithm.

In MSPCA, based on MMECE, a threshold can be set here to select the PCA filter levels. If the decomposition level ($j + 1$) is the first level whose MMECE is larger than this preset threshold, all the coefficients from level 1 to level j will be set to 0, which will reduce the computation of PCA by j times.

3.2.2 Bayesian PCA

The conventional PCA is obtained by computing the eigenvectors of the sample covariance matrix. However, it is difficult to select PC numbers automatically for the

conventional PCA. Based on the probabilistic reformulation of PCA, Bishop introduces a Bayesian treatment of PCA which can determine the effective PC numbers automatically [10].

For a dataset X of N zero-mean data points which are all m dimensional, the latent variable $\mathbf{y}$ is introduced whose prior distribution is a zero mean Gaussian. Thus the observed variable $\mathbf{x}$ can be given as: $\mathbf{x} = \mathbf{V}\mathbf{y} + \boldsymbol{\varepsilon}$, in which $\mathbf{V}$ is the $m \times k$ loading matrix, $\boldsymbol{\varepsilon}$ is a white noise with zero mean and variance $\sigma^2 \mathbf{I}$. Therefore, the marginal distribution of $\mathbf{x}$ should be a zero mean Gaussian distribution, and the likelihood function can be derived as:

$$L(\mathbf{V}, \sigma^2) = -\frac{N}{2} \{m \ln(2\pi) + \ln |\mathbf{C}| + \text{Tr}[\mathbf{C}^{-1} \mathbf{S}]\} \quad (3.3)$$

where $\mathbf{C} = \mathbf{V}\mathbf{V}^T + \sigma^2 \mathbf{I}$ and $\mathbf{S}$ is the covariance matrix of the dataset.

In order to determine the PC number k automatically, the Bayesian treatment is proposed. To implement the Bayesian PCA algorithm, a hyper-parameter vector $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_k\}$ is introduced here to govern the dimensionality of the latent subspace. Therefore, each column of the loading matrix $\mathbf{V}$ is controlled by the corresponding hyper-parameter through this conditional Gaussian distribution:

$$p(\mathbf{V} | \boldsymbol{\alpha}) = \prod \left(\frac{\alpha_i}{2\pi} \right)^{m/2} \exp \left\{ -\frac{1}{2} \alpha_i \|\mathbf{v}_i\|^2 \right\} \quad (3.4)$$

When α_i becomes very large, the corresponding $\mathbf{v}_i$ tends to be zero and can be removed from the loading matrix.

From Bayes' theorem, the log posterior distribution of $\mathbf{V}$ is given by

$$\ln p(\mathbf{V} | D) = L - \frac{1}{2} \sum \alpha_i \|\mathbf{v}_i\|^2 + \text{const} \quad (3.5)$$

where L is defined in (3.3). The solution of $\mathbf{V}$ and σ^2 can be obtained by using the Expectation-Maximization (EM) algorithm to maximize this log posterior distribution.

In the E-step, the evaluation of expected sufficient statistics of the latent variables can be given by

$$E(\mathbf{y}_n) = \mathbf{M}^{-1} \mathbf{V}^T \mathbf{x}_n \quad (3.6)$$

$$E(\mathbf{y}_n \mathbf{y}_n^T) = \sigma^2 \mathbf{M}^{-1} + E(\mathbf{y}_n) E(\mathbf{y}_n)^T \quad (3.7)$$

where $\mathbf{M} = (\mathbf{V}^T \mathbf{V} + \sigma^2 \mathbf{I})$. The updating of parameters in the M-steps is given as:

$$\alpha_i = \frac{m}{\|\mathbf{v}_i\|^2} \quad (3.8)$$

$$\begin{aligned} \tilde{\mathbf{V}} &= [\sum_n \mathbf{x}_n E(\mathbf{y}_n)^T] [\sum_n E(\mathbf{y}_n \mathbf{y}_n^T) + \sigma^2 \mathbf{A}]^{-1} \\ &= \mathbf{S} \mathbf{V} \mathbf{M}^{-1} [\mathbf{M}^{-1} \mathbf{V}^T \mathbf{S} \mathbf{V} \mathbf{M}^{-1} + \sigma^2 \mathbf{M}^{-1} + \sigma^2 \frac{\mathbf{A}}{N}]^{-1} \end{aligned} \quad (3.9)$$

$$\begin{aligned} \tilde{\sigma}^2 &= \frac{1}{Nm} \sum_{n=1}^N \{ \mathbf{x}_n^2 - 2E(\mathbf{y}_n)^T \tilde{\mathbf{V}}^T \mathbf{x}_n + Tr[E(\mathbf{y}_n \mathbf{y}_n^T) \tilde{\mathbf{V}}^T \tilde{\mathbf{V}}] \} \\ &= \frac{1}{m} Tr(\mathbf{S} - \mathbf{S} \mathbf{V} \mathbf{M}^{-1} \tilde{\mathbf{V}}^T) \end{aligned} \quad (3.10)$$

where $\mathbf{A} = diag(\alpha_i)$ and $\mathbf{S}$ is the covariance of sample dataset. Therefore by iterating this formula in the EM algorithm, PC number k can be determined automatically and the loading matrix $\mathbf{V}$ is easy to be computed.

It is worth pointing out that BPCA is much faster than the conventional-PCA. The time complexity of conventional-PCA is $O(m^2n)$, while BPCA's time complexity

is just $O(kmn)$, where n is the number of data points, m is the dimensionality of the dataset, and k is the selected PC numbers. For large m , and $k \ll m$, BPCA is much efficient than the conventional PCA.

Therefore in MSPCA algorithm, this Bayesian-PCA (BPCA) can be utilized to filter the coefficient matrix instead of conventional-PCA as BPCA has the advantages of lower time complexity and automatic selection of PC numbers.

3.2.3 Improved-MSPCA

Aiming at overcoming the two drawbacks of the Conventional-MSPCA, the proposed Improved-MSPCA is shown in Fig. 3.2. The main scheme of Improved-MSPCA is as follows:

Figure 3.2: Procedures of Improved-MSPCA

- *Step 1:* Perform wavelet transform at each column of the data matrix X .
- *Step 2:* Compute the separating level j based on MMECE.
- *Step 3:* For the detail matrices from level 1 to j (high frequency part), set their coefficients to 0.

- *Step 4:* Perform BPCA to filter the coefficients in the approximation and other detail matrices.
- *Step 5:* Compute the inverse wavelet transform based on these simplified detail and approximation matrices.
- *Step 6:* Perform BPCA on the final reconstructed matrix once again to get the normal subspace data $\tilde{X}'$

The Improved-MSPCA separates the low-frequency and high-frequency matrices based on the threshold of MMECE. For the high frequency matrices, all the corresponding wavelet coefficients are set to zero. For the low frequency part, BPCA is utilized to separate the normal and anomalous information in each scale. Moreover, there is no need to choose the number of retained PC for each scale by using BPCA.

Assuming that level j is selected as the tipping point of the decomposition levels, the Improved-MSPCA algorithm can reduce the times of PCA from $(L+2)$ to $(L-j+2)$. In addition, BPCA is adopted to separate the anomalous information rather than applying the conventional-PCA. If for all of the remaining BPCA, the algorithm all selects k PCs ($k \ll m$), the entire time complexity of Improved-MSPCA is $O(mn + (L-j+2)kmn)$, which is much faster than the Conventional-MSPCA.

In conclusion, the proposed Improved-MSPCA is supposed to separate subspaces efficiently with lower time complexity. Meanwhile, this algorithm only requires the setting of a few parameters: mother wavelet, wavelet decomposition level, and the MMECE threshold. In the following evaluations, the algorithm selects "symlet" mother wavelet and sets the decomposition level as $L = \log_2 n - 5$ as illustrated in Section 3.3. The MMECE threshold is determined as 3%, based on Table 3.1.

Figure 3.3: Main Procedures of the Improved-MSPCA Detection System

3.2.4 Improved-MSPCA anomaly detection system

Unlike most of the existing works on PCA-based detection algorithms, the proposed detection system is more focused on multiple flow features instead of detecting the volume changes. For diverse kinds of network attacks nowadays, it is difficult to detect them by only monitoring network volume features. Thus it is necessary to propose the feature-based anomaly detection system to achieve high detection accuracy.

As displayed in Fig. 3.3, the Improved-MSPCA detection system is composed of the following steps:

Step 1: Generation of Network Flow Data As the detection system is mainly based on the detection upon multiple flow-based features, the first step of this detection system is to convert tcpdump data (packet-based) to flow-log data. TCP, UDP, and ICMP network flows are aggregated here.

Step 2: Feature Analysis The entire flow-log data is broken into several time slots and flow-based features are aggregated from each time slot. Empirically, for three main types of flows, the detection system extracts the following features separately: number of flows, some statistical parameters of each flow's packet number (mean, median, and standard deviation), average bytes of flows, average bytes of packets, entropy of IP addresses and ports (source and destination). In this step, the detection system can get a $n \times 30$ data matrix, where n is the number of time slots. As these flow-level features can be calculated in linear time, the time complexity of generating these features is just $O(n)$.

Step 3: Improved-MSPCA based Anomaly Detection For the data matrix X derived in Step 2, Improved-MSPCA is applied to get the normal data $\tilde{X}'$. Therefore, the residual data $\hat{X} = X - \tilde{X}'$ can be obtained. The Square Prediction Error $SPE_i = \|\hat{x}_i\|^2$ is adopted here to visualize the anomalous degree.

Step 4: Control Chart Decision The residual subspace is monitored by plotting the time series of SPE values. Here, Shewhart control chart is utilized to determine which time slot is anomalous. Therefore, the system determines each time slot as:

$$c(x_i) = \begin{cases} normal & SPE_i < \theta \\ anomalous & SPE_i > \theta \end{cases} \quad (3.11)$$

where θ is a threshold which can be tuned in the following evaluation.

It is to be noted that the length of time slot in Step 2 is an important parameter for network anomaly detection. It should depend on the duration of attacks that the system wants to detect. Let us define the quotient $q := t_0/t_1$, where t_0 represents the duration of attacks and t_1 denotes the length of each time slot. If q is too large ($t_0 \gg t_1$), the algorithm may treat the anomalous traffic pattern as a normal trend. If the quotient is too small ($t_0 \ll t_1$), the algorithm may be misguided by the "noise" in the network. Therefore, the length of the time slot is set at the same time level as the duration of attacks.

3.3 Evaluation of Improved MSPCA-based Anomaly Detection System

In order to evaluate the Improved MSPCA-based anomaly detection system, the author uses the 1999 DARPA dataset. This is one of the most remarkably labeled intrusion detection dataset. Some kinds of attacks in this dataset do not influence the traffic patterns [74]. Hence, it is better to apply signature-based detection systems to detect them. Therefore, the evaluation is mainly based on the detection of the DoS attacks.

3.3.1 Detection Results of Improved-MSPCA

In order to evaluate the algorithm, the author utilizes the following metrics:

- *True Positive Rate.* $TP/(TP+FN)$
- *False Positive Rate.* $FP/(TN+FP)$
- *Receiver operating characteristic (ROC) curve.* It is created by plotting true positive rate against false positive rate by selecting different thresholds. Based on the ROC evaluations, the larger area below the ROC curve usually indicates the better performance of detection system.

TP is short for True Positive and FP is for False Positive.

The internal sniffed tcpdump data of DAPRA 1999 dataset is chosen to test the algorithm, as the internal data contains evidence of both internal and external attacks. The DoS attacks in this dataset are almost in the minute-level, so the system aggregates the tcpdump data at 1-minute interval empirically. As DARPA collected data for twenty-two hours a day, the system can construct a 1320×30 time series matrix per

Figure 3.4: ROC curve of Improved-MSPCA detection algorithm

day, where 1320 is the number of total instances for one day and 30 is the feature number. Hence, the Improved-MSPCA algorithm can be applied to these matrices to detect the anomalous time bins of each day.

As mentioned in Section 3.2.3, the "symlet" mother wavelet is adopted to decompose the data to level $L = \log_2 n - 5$. The threshold of MMECE is set as 3%.

After the Improved-MSPCA modeling, the SPE of the traffic data can be calculated. By changing the threshold θ in Shewhart control chart, the ROC curve of this detection algorithm can be plotted. Here, the detection results on Monday, Week 5 are presented as an example. Fig. 3.4 plots the ROC curves of that day, and its corresponding thresholds.

Mother Wavelets with different Vanishing Moments

Here, the author examines the detection results by choosing mother wavelets with different orders of vanishing moments. Fig. 3.5 presents the ROC curves for different

Figure 3.5: Comparison of Improved-MSPCA algorithms for mother wavelet with different orders of vanishing moments

mother wavelets.

For the *sym4*, *sym6*, *sym8* and *sym10*, the detection results are almost the same and better than the other mother wavelets. Although these 4 mother wavelets all give good results, it is better to choose *sym10* so that it can separate the anomalous part more efficiently because of the larger order of vanishing moments [120].

3.3.2 Comparing with Existing Intrusion Detection Methods

Although many researchers have proposed many network anomaly detection systems, it is difficult to conduct a fair comparison among all these different approaches. In this subsection, the author only compares the proposed detection algorithm with three typical detection approaches. The first method is the classical PCA-based detection method introduced by [68]. The second one is a modified version of PCA-based al-

Figure 3.6: Scree plots for PCA and KL_PCA

gorithm, KL-PCA [12]. This approach incorporates the simple PCA with Karhunen-Loeve (KL) expansion, which can deal with the temporal correlation in data as well as the spatial correlation. The last approach is a wavelet-based anomaly detection algorithm [80].

The author implements the first two algorithms and applies them to the DARPA 1999 dataset. PCA and KL-PCA need to select the top_k principal components to construct the normal subspace. According to the analysis in [93], the author utilizes scree plot to determine this top_k parameter for each method. As seen from Fig. 3.6, the scree plot of PCA has the knee in the range of [5, 7], while KL-PCA is in the range of [22, 28]. Thus the author determines the top_k value of 6 for PCA-based detection method and 26 for KL-PCA detection algorithm. Another important parameter in KL-PCA is the number N of the time shift. As discussed in [12], the author sets N as 3 so as to get the best detection accuracy. For the wavelet-based algorithm, the results are from [80].

Results of Different Detection Algorithms

The author compares PCA, KL-PCA, conventional-MSPCA approaches with

Figure 3.7: ROC curves for PCA-based, KL-PCA-based and MSPCA-based detection algorithm

Improved-MSPCA in the aspect of ROC curves. Fig. 3.7 presents the ROC curves of PCA, KL-PCA, conventional-MSPCA and Improved-MSPCA. These ROC curves show that conventional-MSPCA and Improved-MSPCA has the best performance among the three PCA-like anomaly detection algorithms. It is observed that conventional-MSPCA and Improved-MSPCA have nearly the same detection performance. The results clearly show that the introduction of MMECE and Bayesian-PCA does not compromise the detection accuracy of MSPCA.

Table 3.2 presents the execution time comparison of PCA, conventional-MSPCA and Improved-MSPCA. The results clearly show that the Improved-MSPCA significantly reduces the execution time of conventional-MSPCA.

Table 3.2: Execution Time Comparison

Algorithm	PCA	conventional-MSPCA	Improved-MSPCA
Running time (s)	0.1069	0.6369	0.1624

Table 3.3 presents a quick quantitative comparison of these detection systems. By

Table 3.3: Detection Results of MSPCA, PCA and wavelet-based algorithm

Attack Type (DoS Attack)	Improved- MSPCA ($\theta = 10.6$)	PCA ($\theta = 3.3$)	KL-PCA ($\theta = 3.5$)	Wavelet-based method	Ground Truth
Apache2	20	16	22	29	30
Neptune	7	6	4	5	7
Pod	5	4	4	1	5
Selfping	4	1	3	2	4
Smurf	1	1	1	2	2
Udpstorm	16	9	6	16	16
Dosnuke	17	4	7	2	17
Syslogd	1	1	1	1	1
Warezcilent	0	1	1	1	1
Crashiis	3	3	3	1	3
Total detected	74	46	52	60	86
Total Alarms	219	221	219	800+	NA

selecting appropriate thresholds for Improved-MSPCA, PCA and KL-PCA, the detection systems can derive 219, 221 and 219 alarms respectively. As revealed by the table, all the methods can detect most DoS attacks. However, MSPCA detects 28 more anomaly instances than PCA and 22 more attacks than KL-PCA.

The last but one column is the results from [80]. Although this wavelet based approach has better performance for certain DoS attacks compared to the proposed MSPCA approach, it generates many false alarms which cause a high false positive rate. It is because, this method analyzes the traffic features separately and for each single feature, the detection accuracy is very low. Therefore, they need to combine the results of different features, which then results in a high false positive rate.

Comparison of Normal Subspace Contamination

The DARPA 1999 dataset is used here to present the comparison between PCA, KL-PCA and Improved-MSPCA in the aspect of normal subspace contamination. The results are presented in Fig. 3.8.

Fig. 3.8 shows the $L2$ norm (SPE) of the normal subspace traffic. These three figures of Fig. 3.8 are for the normal subspaces constructed by PCA, KL-PCA and Improved-PCA approaches. The spikes in Fig. 3.8(b) and Fig. 3.8(c), which have been circled, show the normal subspace contamination of the PCA and KL-PCA methods. Compared to the Fig. 3.8(b), the magnitudes in Fig. 3.8(a) are much smaller and have no spikes.

As illustrated in Fig. 3.1 and Fig. 3.2, MSPCA builds PCA model at each time and frequency scale of the original dataset. Thus MSPCA can separate normal and anomalous information at any time and frequency scale, whereas the conventional PCA method just separates the anomalous information in a single scale. Therefore, MSPCA performs better when facing normal subspace contamination.

3.3.3 Conclusion on Improved-MSPCA algorithm

Due to the problem of normal data contamination, some particular network attacks e.g. Udpstorm and Dosnuke attacks cannot be detected by the PCA algorithm. The Conventional-MSPCA algorithm combines PCA with wavelet analysis so that it can separate the normal and anomalous data in different time and frequency scales. Thus, these hidden attacks can be detected easily by the Conventional-MSPCA. Furthermore, the Improved-MSPCA algorithm requires much less parameters resulting in a lower time complexity compared to the Conventional-MSPCA algorithm.

3.4 Sketch-MSPCA Anomaly Detection System

The last section reveals the great improvement by applying Improve-MSPCA to the network anomaly detection. However as mentioned in [93], another limitation of PCA-

Figure 3.8: Normal Subspace Contamination of two algorithms

based detection algorithms is that they fail to tell which IP address is anomalous. In [60, 61, 71], they proposed a novel system which combined the sketch data structure and PCA-based detection to detect the anomalous IP addresses.

In this section, the author introduces the sketch data structure and proposes the novel anomaly detection system which combines the sketch data structure and Improved-MSPCA based anomaly detection algorithm. The incorporation of sketch data structure is primarily to compactly store the voluminous network flows in an efficient manner to lend itself to efficient processing. Meanwhile, this combination enables the anomalous IP addresses to be identified.

3.4.1 Sketch Data Structure

Sketch is a compact summary data structure which is capable of storing high-dimensional measurement data. One sketch table comprises M rows with N sketches each, which results in a $M \times N$ array $H[l][d]$. Each row l of these tables is associated with a particular hash function h_l . Thus, for each hash function h_l , the incoming data with the hash key key_i will be updated to the corresponding data block $(l, h_l(key_i))$. The updating procedure is shown in Fig. 3.9.

Figure 3.9: The updating procedures of sketch structure

According to [35], the 4-universal family of hash functions [111] is a good choice to avoid the collisions between different hash keys. In this thesis, the author adopts the following 4-universal hash functions:

$$h(x) = \sum_{i=0}^3 a_i x^i \bmod p \bmod N \quad (3.12)$$

where p is a random prime and a_i is picked randomly from $[0, p - 1]$. In order to get a fast implementation of 'mod p ', the author uses p as $2^{61} - 1$ which is a Mersenne prime.

In this detection system, the author selects the source IP address as the hash key to split the network trace.

The author applies detection algorithm for each sub-traffic and detects suspicious

time bins. Since anomalies have been randomly shuffled through different hash entries, anomalous source IP addresses can be identified by the approximate agreements among different sketches.

3.4.2 Detection System Architecture

The proposed detection system utilizes sketch data structure to split the raw network traces, and then adopts the Improved-MSPCA to detect the anomalous time bins of each sub-traces. As revealed in Fig. 3.10, the detailed procedures of the Sketch-MSPCA detection system are composed of the following:

Figure 3.10: Sketch-MSPCA Anomaly Detection System

Step 1: Generation of Network Traffic Data Just like the Improved-MSPCA detection algorithm, this sketch-MSPCA anomaly detection mainly analyzes the flow-based features. Thus, the first step is also to transform the raw data. The detection system still aggregates the network flows of the three dominant protocols: TCP, UDP, and ICMP.

Step 2: Sketch Construction The sketch data structure is used here to split the original traffic flows into several sub-traffic traces. For each incoming traffic $\{srcIP_i, flow_i\}$, this flow will be allocated to the sub-traffic trace d according to the hash value of the flow's source IP address $\{d = h_l(srcIP_i), l \in 1, \dots, M, d \in 1, \dots, N\}_{l,d}$. As a result, for each hash function h_l , the original network trace is split into N sub-traffic traces.

Step 3: Flow-based Features Generation For each sub-trace, the system aggregates the traffic into an amount of time bins and extract the flow-based features from

each time slot. Just like the Improve-MSPCA detection system, the following features are extracted: number of flows, some statistical parameters of each flow's packet number (mean, median, and standard deviation), average bytes of flows, average bytes of packets, entropy of IP addresses and ports (source and destination). The changes of these features may reflect different kinds of network attacks. These metrics are extracted separately for three different types of flows derived from Step 1.

Step 4: MSPCA-based Anomaly Detection In this step, the proposed Improved-MSPCA is applied to each data matrix X to separate the normal data $\tilde{X}'$ and residual data $\hat{X}$. For each time slot i , the system can derive the Square Prediction Error ($SPE_i = \|\hat{x}_i\|^2$), which can be utilized here to model the abnormal changes in this residual matrix. Thus it is able to get the information of suspicious time points based on the Control Chart evaluation of SPE. For each sub-trace, the source IP addresses in the flagged anomalous time slots will be recorded; and for each hash function, the anomaly results of its corresponding sub-traces will be aggregated into a list. For the reason that M hash functions are being used, this step will give rise to M lists of anomalous source IP addresses

Step 5: Anomaly Identification The system intersects these anomalous source IP address list to identify the anomalous source IP addresses. The source IP address will be determined as an anomaly if its number of alarms exceeds $(M/2 - 1)$.

In step 4, Median Absolute Deviation (MAD) is applied to monitor the abnormal changes in the residual data of each sub-trace, as MAD is more robust to the distribution of residual data [5]. Therefore, the anomalous time slots will be determined if $SPE > Median + \theta \times MAD$. θ is a parameter which can be tuned to get different detection performances.

The proposed detection system needs to perform MSPCA to each sub-trace in each

Table 3.4: Parameter Settings of Sketch-MSPCA System

Hash Number	8
Mother Wavelet	Sym10
Decomposition level	$\log_2 n - 5$
MMECE Threshold	3%

sketch. Assuming that the system takes i sketches into account and splits the traffic into j sub-traces for each sketch, the total computational time of Sketch-MSPCA system will be ij times compared to MSPCA's $O(mn + (L - j + 2)kmn)$.

3.5 Evaluation of Sketch-MSPCA system

In this section, the author will give the evaluation results of the proposed Sketch-MSPCA system based on some famous dataset. In the system, some parameters need to be preset here. 8 hash functions are selected at the beginning, as this is enough to obtain a correct identification of anomalous source IP addresses, according to the study of [35]. As discussed in Section 3.3.1, for the Improved-MSPCA anomaly detection algorithm in Step 4, "sym10" mother wavelet is chosen to decompose the data and the decomposition level is $L = \log_2 n - 5$, where n is the number of time bins. The MMECE threshold is determined as 3%. These preset parameters are listed in Table 3.4

3.5.1 MAWI Dataset

In order to evaluate this Sketch-MSPCA detection system, the author has used the backbone traffic traces from the Measurement and Analysis of the WIDE Internet (MAWI) working group [45]. This group has been providing raw traffic traces for nearly 15 years (2001 - 2015). The author randomly selects nine days of data from

Table 3.5: Detail Information of Network Traces of three days in 2013

Evaluation Date	2013.04.20	2013.07.13	2013.09.21
Number of Source IP addresses	3771951	4175694	4295471
Number of Total Packets	46006144 (100.00%)	40245170 (100.00%)	50051873 (100.00%)
Number of Total TCP Packets	22136298 (48.12%)	14100943 (35.04%)	19856920 (39.67%)
Number of Total UDP Packets	2245093 (4.88%)	1793355 (4.46%)	6034877 (12.06%)
Number of Total ICMP Packets	19906350 (43.27%)	21241464 (52.78%)	22170284 (44.29%)

Table 3.6: Detail Information of Network Traces of three days in 2014

Evaluation Date	2014.02.22	2014.06.14	2014.09.10
Number of Source IP addresses	440316	13853659	9016292
Number of Total Packets	86708170 (100.00%)	106536906 (100.00%)	92953331 (100.00%)
Number of Total TCP Packets	54237021 (62.55%)	42200344 (39.61%)	50988063 (54.85%)
Number of Total UDP Packets	3375369 (3.89%)	2445394 (2.30%)	1754331 (1.89%)
Number of Total ICMP Packets	21714321 (25.04%)	56100735 (52.66%)	33995848 (36.57%)

2013, 2014 and 2015. The detail information of these nine days of network traces are presented in Table 3.5, 3.6, 3.7. As seen from the tables, for these 9 days of data, TCP, UDP, and ICMP packets nearly occupy the entire traces. That is why the author only analyzes the features of these three types of network flows. For MAWI dataset, the data traffics are aggregated at 1 second interval. Empirically, the author believes this is reasonable as the attack whose duration is smaller than 1 second is unlikely to cause significant damage.

Relying on the heuristics used in [11, 52], the author labels the normal and anomalous source IP addresses, which then serve as the ground truth for further evaluations.

Table 3.7: Detail Information of Network Traces of three days in 2015

Evaluation Date	2015.01.03	2015.03.04	2015.05.02
Number of Source IP addresses	11557446	5063104	4901419
Number of Total Packets	90666867 (100.00%)	92575975 (100.00%)	78965271 (100.00%)
Number of Total TCP Packets	35234991 (38.86%)	59173952 (63.92%)	41676003 (52.78%)
Number of Total UDP Packets	6495426 (7.16%)	2802067 (3.03%)	6814548 (8.63%)
Number of Total ICMP Packets	46146455 (50.90%)	22544145 (24.35%)	24470818 (30.99%)

The author evaluates these systems based on the three criteria: True Positive Rate (TPR), False Positive Rate (FPR) and ROC curve. In this evaluation, TPR is defined as Total number of anomalous source IP addresses identified by the system divided by total number of attack source IP addresses labeled by the heuristics. FPR is defined as total number of normal source IP addresses identified by the system divided by total number of non-attack source IP addresses labeled by the heuristics. The ROC curve is created by plotting True Positive Rate against False Positive Rate at different threshold settings.

3.5.2 Evaluation of Different Sketch Sizes

Except the preset parameters listed in Table 3.4, the proposed system still needs to tune the sketch size (N) in Step 2. In this subsection, the author explores the parameter space based on the analysis of MAWI network traces.

Sketch size is also the number of sub-traffic traces for each hash function. The author evaluates the performance of system by setting the sketch size as 8, 12, 16, 24 and 32. By changing θ in Step 4, the author can derive different TPR and FPR and plot the ROC curves. By comparing the ROC curves of these five situations, one can

obtain the effects of different sketch sizes.

The author just presents the results of 2015.05.02, but note that the detection results of other days are quite similar. Fig. 3.11 gives the *ROC* curves of the detection results under different sketch sizes. As seen from the results, the 32-sketches system has the best performance among these four situations. If the sketch size is too small, the sketch data structure is useless as it fails to reveal the hidden anomalies. That is why the system with small sketch size has worse performance. However, when the system attempts to split the network traces into 64 sub-traffic traces (64 sketches), the Improved-MSPCA cannot be performed in some sub-traffic traces. Too large a sketch size may lead to sparse sub-traces which will result in some sparse feature matrices. That is why MSPCA model cannot be built on these matrices.

Figure 3.11: ROC curves of different sketch numbers for the data on 2015.05.02

In conclusion, for the following evaluations, the author decides to choose the sketch size of 32, as it is an appropriate sketch size which can split the network traffic properly and give a good result.

3.5.3 Comparison with other Sketch-based Detection Systems

To show the effectiveness of the proposed system, it is necessary to compare the proposed Sketch-MSPCA detection system with the other sketch-based detection systems. One similar system is proposed in [60,61], which combines the two hash-divides (sketches) with PCA detection algorithm. The first hash-divide is meant for determining the anomalous source IP addresses. The second hash-divide is to build the PCA model. In [61], they collect packet counting in each sub-traffic trace to build the PCA model, whereas, in [60], they compute the packet entropy of each sub-traffic trace. The author implements these the systems from these two papers and they will be referred to as "PACKET-PCA" and "ENTROPY-PCA" for convenience.

There are two main differences between their systems and the proposed system. One is that the proposed system collects multiple flow-based features so that there is no need to perform the second hash-divide. The other major difference is that the proposed system adopts Improved-MSPCA to detect anomalies in each sub-traffic. Thus, the proposed system will also be named as "FLOW-MSPCA" in the following evaluations.

As "PACKET-PCA" and "ENTROPY-PCA" have the same detection mechanism, the author just set the same parameters for them. Based on the analysis of [61] and [60], 8 hash functions are selected to decompose the dataset and the sketch size of these two systems are set as 8×8 . As these two systems both apply PCA to detect anomalies in each sub-traffic, it is important to determine the number of principal components in these systems. For all the sub-traffics, the author decides on the top_k principal components based on the cumulative percent variance captured (CPV). CPV is empirically set as 0.7 based on the results in [93] [61]. For the proposed FLOW-MSPCA system, the author selects 8 hash functions and sets the sketch size as 32 as

Table 3.8: Parameters of three detection systems

Detection System	No of Hash Function	Size of First Sketch	Size of Second Sketch	Threshold of CPV
PACKT-PCA	8	8	8	0.7
ENTROPY-PCA	8	8	8	0.7
FLOW-MSPCA	8	32	NA	NA

previously mentioned. All the parameters of the three systems are listed in Table 3.8.

In order to compare these three systems, the author first applies the data of 2015.05.02 to test them. Fig. 3.12 presents the *ROC* curves of these three systems. Apparently, ENTROPY-PCA has a small improvement compared to PACKET-PCA. However, the proposed system outperforms the other two systems. The results of FLOW-MSPCA shows that the area under its *ROC* plot is much larger than the other two systems. For similar false positive rate, the true positive rate for FLOW-MSPCA has increased about 10%, which is a significant improvement compared to the other two systems.

Figure 3.12: ROC curves of three detection systems for the data on 2015.05.02

To ensure this is not a special case for the proposed system, the author has investigated the detection results of other days' network traces which show similar results. Based on the *ROC* curves in Fig. 3.12, the thresholds can be selected separately for these three systems. Therefore, the detection results of the other 8 days can be derived by using these fixed thresholds.

Since the characteristics of traffic vary everyday, the optimal thresholds also fluctuate according to the corresponding network traces. Based on the ROC curves of 2015.05.04, the author selects the thresholds of 3.9, 1.7 and 2.1 for PACKET-PCA, ENTROPY-PCA, and FLOW-MSPCA respectively. The author evaluates the other days of data using these fixed thresholds so that their robustness to the traffic fluctuations can be verified.

The detailed results of these three systems are listed in Table 3.9, 3.10, 3.11. Apparently, FLOW-MSPCA performs better than the other two systems. For every day's data, FLOW-MSPCA detects more attacks than the other two systems. On the whole, FLOW-MSPCA detects 85 more attacks than ENTROPY-PCA and 107 more attacks than PACKET-PCA. These overall results indicate that FLOW-MSPCA outperforms PACKET-PCA and ENTROPY-PCA. This improvement stems from two aspects: the proposed system utilizes flow-based features to reveal the hidden anomalies and adopts Improved-MSPCA to detect the anomalous time bins.

It is also important to investigate the difference in attack types detected by PACKET-PCA, ENTROPY-PCA, and FLOW-MSPCA. According to the heuristics [11, 52], MAWI classifies the attacks into 17 categories. Table. 3.12 presents the detection results of different attack types by using these three systems. FLOW-MSPCA has the best detection performance when facing each kind of attacks in MAWI dataset. Especially facing SYN attacks, PACKET-PCA and ENTROPY-PCA can only detect half of these attacks, whereas FLOW-MSPCA detects nearly all this kind of attacks.

Table 3.9: Detection Results of Packet-PCA
 Packet-PCA
 $(\theta = 3.9)$

Date	Attacks	Non-Attacks	TPR	FPR
2013.04.20	43	6572	97.727%	0.174%
2013.07.13	73	4407	90.123%	0.106%
2013.09.21	104	4528	71.727%	0.105%
2014.02.22	34	6966	58.621%	0.157%
2014.06.14	62	6946	78.841%	0.050%
2014.09.10	55	6260	63.953%	0.069%
2014.09.10	115	9851	80.851%	0.195%
2015.05.02	59	6287	84.286%	0.128%
2015.09.11	69	9192	88.462%	0.180%
Total Results	613	61009	78.389%	0.112%

Table 3.10: Detection Results of Entropy-PCA
 Entropy-PCA
 $(\theta = 1.7)$

Date	Attacks	Non-Attacks	TPR	FPR
2013.04.20	43	5068	97.727%	0.134%
2013.07.13	74	2722	91.358%	0.065%
2013.09.21	121	3337	83.448%	0.078%
2014.02.22	40	7893	68.966%	0.177%
2014.06.14	59	5151	78.841%	0.037%
2014.09.10	55	5443	63.953%	0.060%
2014.09.10	112	9944	80.851%	0.196%
2015.05.02	61	5724	84.286%	0.117%
2015.09.11	70	9056	88.462%	0.178%
Total Results	635	54338	78.389%	0.099%

Table 3.11: Detection Results of Flow-MSPCA
($\theta = 2.1$)

	Flow-MSPCA ($\theta = 2.1$)			
Date	Attacks	Non-Attacks	TPR	FPR
2013.04.20	44	5494	100.000%	0.146%
2013.07.13	76	3769	93.827%	0.090%
2013.09.21	136	3802	93.793%	0.089%
2014.02.22	55	8610	94.828%	0.194%
2014.06.14	73	6461	92.405%	0.047%
2014.09.10	76	6631	88.372%	0.074%
2014.09.10	120	9251	85.106%	0.183%
2015.05.02	67	5670	95.714%	0.116%
2015.09.11	73	11657	93.590%	0.229%
Total Results	720	61345	92.072%	0.112%

Table 3.12: Types of attacks detected by three systems

Attack Types		PACKET PCA	ENTROPY PCA	FLOW MSPCA	Ground Truth
Attacks from well known suspicious ports	Sasser worm	0	0	0	0
	Netbios attack	2	2	2	2
	RPC attack	0	0	0	0
	SMB attack	0	0	0	0
	SYN attack	102	140	202	230
	RST attack	2	1	2	2
	FIN attack	0	0	0	0
	Ping flood	6	4	8	8
	FTP attack	0	0	0	0
	SSH attack	0	0	0	0
	HTTP attack	2	2	2	2
	HTTPS attack	2	2	2	2
Attacks from other well known ports	FTP attack	0	0	0	0
	SSH attack	19	19	19	19
	HTTP attack	210	208	212	228
	HTTPS attack	118	116	119	128
	Unknow ports attack	150	141	152	161
Total Numbers of Attacks		613	635	720	782

Table 3.13: Anomaly Degrees for the Selected Anomalous Source IP Addresses

Date	Source IP Address	Anomaly Type	Anomaly Score
2013.04.20	153.22.12.225	TCP_RST_ACK_response	77.73
	210.196.184.226	network_scan_SYN_t	55.01
	210.204.226.44	network_scan_SYN_t	54.79
	199.99.113.121	multipoint_to_multipoint	54.09
	198.96.194.29	multipoint_to_point_HTTP	52.00
2014.06.14	175.124.175.7	TCP_RST_ACK_response	80.00
	107.41.66.68	network_scan_SYN_t	45.40
	204.102.131.84	multipoint_to_multipoint	43.19
	204.106.164.147	network_scan_ACK	37.19
	107.41.66.8	network_scan_SYN_t	36.18
2015.03.04	13.166.1.133	network_scan_SYN_t	64.99
	184.32.128.107	TCP_RST_ACK_response	53.25
	143.3.253.10	multipoint_to_multipoint	51.40
	103.161.118.168	network_scan_ICMP_eqrq	51.08
	143.3.179.159	network_scan_SYN_t	48.23

3.5.4 Anomaly Degrees for the Detected Attacks

For PCA-based anomaly detection system, the anomalous points can be determined by thresholding the Square Prediction Error (SPE). This SPE value can also provide us anomaly degrees of these anomalous points. Therefore, by modifying Steps 4 and 5 of Sketch-MSPCA detection system in Section 3.4.2, the system can also report the anomaly degree of detected IP addresses. The first 3 steps remain unchanged, while Steps 4 and 5 will be changed as follows:

- **Step 4: MSPCA-based Anomaly Detection** After determining the anomalous time points, the system will record not only the source IP addresses, but also the SPE values of their corresponding time slots. Therefore, for each hash function, the system can derive a dictionary which stores the suspicious source IP addresses and its related SPE values.
- **Step 5: Anomaly Identification** The source IP address will be determined as an anomaly if more than half of the dictionaries report it. Thus the anomaly

degree of this source IP address is derived by summing up its corresponding SPE values in these reported dictionaries.

Therefore, the proposed Sketch-MSPCA detection system is able to generate a list which consists of detected anomalous source IP addresses and their corresponding anomaly degrees. Obviously, the larger the anomaly degree, the more probable is the attack. Table 3.13 presents some selected anomalous source IP addresses and their corresponding anomaly degrees.

For incoming network traffic where one does not know whether there are any attacks, the Sketch-MSPCA detection system can be applied to get the potential anomalous source IP addresses list. The derived anomaly degree will then be able to provide insights into how probable a source IP address could be a potential source of an attack.

3.6 Conclusion

In this chapter, the author has proposed an Improved-MSPCA anomaly detection algorithm. By introducing the MMECE threshold and Bayesian-PCA, Improved-MSPCA overcomes two limitations of the Conventional-MSPCA and can be applied to different network scenarios easily. Meanwhile, the author presents a novel feature-based detection system which is a combination of Improved-MSPCA detection algorithm and Sketch data structure.

By evaluating on DARPA 1999 dataset, large improvements in the Conventional-MSPCA and the Improved-MSPCA algorithms can be observed compared to the other three detection algorithms in terms of higher true positive rate and lower false positive rate. In addition, the Improved-MSPCA algorithm reduces the execution time when compared to the Conventional-MSPCA. The author also evaluates the proposed

Sketch-MSPCA detection system on MAWI dataset. It can be confirmed that the proposed system is superior to the PACKET-PCA and ENTROPY-PCA systems due to the generation of flow-based features and the usage of Improved-MSPCA detection algorithm.

Chapter 4

Power Spectrum Entropy based Detection and Mitigation of Low-Rate DoS Attacks

In this chapter, the author focuses on the detection and mitigation of Low-Rate Denial-of-Service (LDoS) attack. LDoS attack is a special class of DoS attacks [64], [82]. This kind of DoS attack exploits TCP's congestion control mechanisms to degrade TCP's throughput. By sending periodically pulsing packets to the bottleneck routers, LDoS attacks can make these routers drop the normal TCP packets so as to decrease the congestion window size of TCP senders. As LDoS attack targets at the bottleneck routers, the network side NIDS may fail to detect this attack.

In order to detect the LDoS attacks, the author introduces a novel detection metric: Power Spectrum Entropy (*PSE*), which combines Power Spectrum Analysis with Information Entropy. The author goes on to propose a LDoS attack mitigation algorithm which incorporates the detection metrics with Robust Random Early Detection algorithm. This algorithm can be applied to the routers so as to detect and mitigate

the LDoS attacks in real time.

The main contributions of this chapter are threefold. Firstly, the author introduces a novel LDoS detection metric which combines Power Spectrum Analysis with Information Entropy. By evaluating on NS-3 simulations and real network traces, this novel detection metric is shown to have higher true positive rate and lower false positive rate than the other existing LDoS detection methods. Secondly, the author proposes a Power Spectrum Entropy-based Robust-RED (PRRED) queuing algorithm which utilizes Power Spectrum Entropy to determine the LDoS attack flows. Thirdly, the author implements the PRRED algorithm in NS-3 so as to compare the LDoS mitigation performance of the proposed algorithm with other related queuing algorithms. The simulation results reveal that PRRED algorithm has a better mitigation performance compared to the traditional queuing algorithms.

4.1 Characteristics of Low-Rate DoS (LDoS) Attacks

Low-Rate DoS (LDoS) attack exploits TCP's congestion control mechanisms, which makes it much harder to be detected compared to the conventional DoS attacks. There are two main congestion control mechanisms which are commonly used in TCP transmission. The first one is triggered by retransmission timeout (RTO) [89]. When this timeout occurs, TCP sender will reduce its congestion window (CWND) to one packet and retransmit that lost packet. The second mechanism is triggered by receiving three duplicate acknowledgments (ACKs). When faced with this packet loss, TCP sender will halve its CWND and enter the fast recovery state to retransmit the lost packets. The second mechanism is generally called an Additive-Increase, Multiplicative-Decrease (AIMD) algorithm.

As illustrated in [65], a LDoS attacker sends periodic packet bursts to the bottleneck

routers to flood them so as to drop normal TCP packets periodically. Based on these periodic packet drops, a LDoS attack can make a normal TCP sender receive some false congestion signals periodically. The normal TCP sender will be forced to reduce its CWND to limit its packet transmission rate resulting in serious degradation of the overall Quality of Service (QoS).

Figure 4.1: Low-Rate DoS Attack

A LDoS attack is normally characterized by three main parameters i.e. (R, D, T) . As presented in Fig. 4.1, R is the attack rate which should be large enough to induce loss. Normally, the attacker will set R equal to the bottleneck linkrate. D represents the duration of the attack burst for every period. T represents the period of LDoS attack. Depending on which congestion control mechanism the attack evokes, the LDoS attack can be classified into *Timeout-based attacks* and *AIMD-based attacks* accordingly [82].

4.1.1 Timeout-based LDoS Attacks

A TCP sender normally sets its RTO according to [89]:

$$RTO = \max(RTO_{min}, SRTT + \max(G, 4RTTV)) \quad (4.1)$$

where G represents the clock granularity, $SRTT$ is the smoothed round trip time and $VRTT$ is the variation of round trip time. RTO_{min} is the lower bound of RTO, which is recommended to be set as 1 second to avoid spurious retransmissions [3]. If $RTO_{min} > SRTT + 4VRTT$, the TCP flows will have equivalent values of RTO. If packet loss happens, TCP sender waits for this RTO to expire. Then it reduces its CWND to one packet and retransmits the lost packet.

When the Timeout-based LDoS attack begins, the victim TCP senders will experience timeout state every T seconds. The value RTO is very likely to be given by RTO_{min} continuously. Assuming that the value of RTO is given by RTO_{min} , the normalized throughput of the normal TCP is given by [65]:

$$\rho(T) = \frac{\left\lceil \frac{RTO_{min}}{T} \right\rceil T - RTO_{min}}{\left\lceil \frac{RTO_{min}}{T} \right\rceil T} \quad (4.2)$$

Therefore, when $T = RTO_{min}$ and $T = RTO_{min}/2$, $\rho(T) = 0$.

4.1.2 AIMD-based LDoS Attacks

Rather than focusing on Timeouts, AIMD-based LDoS attacks exploit the TCP's AIMD Mechanism. When three duplicate acknowledgments arrive at the TCP sender, TCP will halve its congestion window and enter the fast-recovery state. While AIMD-based LDoS attacks may not cause the CWND to be reduced to 1, they can still significantly degrade the QoS.

Fig. 4.2 depicts the resulting TCP congestion window changes when counting LDoS attacks. It represents a saw-tooth pattern for CWND changes. By ignoring short fast recovery phase, [26] describes a simplified analytical model for TCP performance degradation. When counting LDoS attacks, the average congestion window size of the

TCP sender is given by:

$$W_{avg} = \frac{3T}{4RTT} \quad (4.3)$$

where T is the period of LDoS attacks and RTT is the round trip time. Thus, the overall throughput can be calculated as:

$$\rho(T) = \frac{W_{avg} \times \frac{T}{RTT}}{T} = \frac{3T}{4RTT^2} \quad (4.4)$$

Figure 4.2: AIMD-based LDoS Attacks

Although LDoS attacks can be roughly classified into these two main categories, these two kinds of attacks may happen together in the real network. In addition, as the LDoS attack just sends periodic pulses, the overall attacking rate is so small that it can easily elude most traditional DoS attack detection mechanisms. Therefore, how to detect and mitigate LDoS attacks become a very important and challenging issue for network security.

4.2 Power Spectrum Entropy based Detection Metrics

In this section, the author combines the Power Spectrum Analysis with Information Entropy to differentiate LDoS attack flows. For the periodic LDoS flows, their Power Spectrum Entropy (PSE) should be smaller compared to that of normal TCP flows. Based on Fourier Analysis and Wavelet Analysis, the author introduces two kinds of *PSE*: Fourier Power Spectrum Entropy (FPSE) and Wavelet Power Spectrum Entropy (WPSE).

4.2.1 Fourier Power Spectrum Entropy (FPSE)

By modeling LDoS attack as the periodic square waves, its Fourier Series Coefficient can be calculated:

$$F_n = \frac{1}{T} \int_0^T f(t) e^{-j2\pi n f_0 t} dt = \frac{RD}{T} e^{-jn\pi f_0 D} Sa(n\pi f_0 D) \quad (4.5)$$

where $Sa(x) = \sin(x)/x$ and $f_0 = 1/T$ is the fundamental frequency. Therefore, the Fourier Power Spectrum Density (PSD) of periodic square waves can be calculated as follows:

$$P(f) = \sum_n \left(\frac{RD}{T} \right)^2 Sa^2(n\pi f_0 D) \delta(f - n f_0) \quad (4.6)$$

$\delta(x)$ is the unit impulse function. Fig. 4.3 depicts the PSD of the periodic square waves, in which power is mostly concentrated at low-frequencies.

In the real network, the packet arrival numbers can be sampled at the router to generate the discrete signal $x(N)$. Thus, PSD of each signal can be calculated by using Discrete Fourier Transform (DFT). Based on the DFT coefficients ($X(k) =$

Figure 4.3: Theoretical results of PSD for the periodic square wave

$\sum_n x(n)e^{-j2\pi kn/N}$, PSD can be easily estimated by:

$$S_x(k) = |X(k)|^2 = \left| \sum_n x(n)e^{-j2\pi kn/N} \right|^2 \quad (4.7)$$

In order to reveal the difference in the LDoS flows and normal TCP flows, the author builds a dumbbell network topology in NS-3 simulator. The detail information of the dumbbell topology is given in Section 4.3.1. A flow is defined by a 3-tuple (Source IP, Destination IP, Protocol). For the normal TCP flows, the author only sets one normal node to communicate with the server by using TCP-reno. For the LDoS attack flows, the author sets one attack node to send LDoS attack packets. The parameters for the LDoS attack are $T = 1.35s$, $R = 10Mbps$, $D = 100ms$. The packet sizes for normal TCP flows and LDoS attack flows are 1500 bytes and 50 bytes respectively. In order to generate the discrete signals, the author samples the packet arrival numbers at a sampling period of $1ms$ at the bottleneck router. For each flow, the time series data

are extracted with a length of 2s.

Thus, PSD for each signal can be calculated according to Eq. 4.7. Fig. 4.4 and Fig. 4.5 plot the discrete signals and PSD estimates for LDoS attacks and normal TCP flows respectively. For LDoS attacks, the resulting PSD estimate is quite similar to the theoretical analysis in Eq. 4.6, whose power is mostly distributed across the very low frequency range. For normal TCP flows, the resulting PSD estimate has a very flat power distribution over a wider frequency range. Therefore, Power Spectrum Density has the ability to reveal the periodicity embedded inside the discrete signals.

Figure 4.4: Discrete signals for the LDoS attack flow and its PSD estimates

Figure 4.5: Discrete signals for the normal TCP flow and its PSD estimates

In order to characterize this difference in these two PSD distributions, the author proposes a novel detection metric: *FPSE*, which combines PSD with Information

Entropy. Information Entropy is able to evaluate different distributions [100]. Distributions which are sharply peaked around a few values (i.e. the distribution of LDoS attacks' PSD) will have a relatively low entropy, whereas those that are spread more evenly across many values (i.e. the distribution of the normal TCP flows' PSD) will have a higher entropy.

Therefore, by combining PSD with Information Entropy, *FPSE* is able to identify the difference in the PSD distributions of LDoS attack flows and normal TCP flows.

In order to calculate *FPSE*, the normalized PSD needs to be normalized first:

$$p(k) = \frac{S_x(k)}{\sum_k S_x(k)} \quad (4.8)$$

Thus, *FPSE* is defined as follows:

$$FPSE = - \sum_k p(k) \log p(k) \quad (4.9)$$

In Section 4.3.2, based on extensive simulations in NS-3, the author will further analyze the distributions of *FPSE* for normal and attack flows. Based on their different distributions, a *FPSE* threshold can be set to determine the LDoS attack flows.

4.2.2 Wavelet Power Spectrum Entropy

In this subsection, the author proposes another LDoS detection metric based on wavelet power spectrum analysis. At the point of research, it is the first time that wavelet power spectrum analysis is applied to detect LDoS attacks.

Wavelet is a set of basic functions that oscillate at both time and frequency domains. Based on this property, wavelet analysis can decompose the original signal into the time

and frequency domains. By applying multiresolution analysis, Mallat [83] introduces a fast algorithm to compute Discrete Wavelet Transform (DWT) using wavelet function $\varphi_{j,k}(t)$ and scale function $\phi_{j,k}(t)$. Thus, the original signal ($x(t)$) will be decomposed into the approximation and detail parts:

$$x(t) = \sum_k c_x(J, k) \phi_{J,k}(t) + \sum_j \sum_k d_x(j, k) \varphi_{j,k}(t) \quad (4.10)$$

where $d_x(j, k)$ is the wavelet coefficient of $x(t)$. Given that the frequency of the original signal ranges from 0 to f , the wavelet coefficients of scale j covers the frequency band from $f/2^{j-1}$ to $f/2^j$.

Based on these wavelet coefficients, the wavelet energy of signal $x(t)$ at scale j can be defined as:

$$E_j = \frac{1}{D_j} \sum_k |d_x(j, k)|^2 \quad (4.11)$$

where D_j is the number of wavelet coefficients at the corresponding decomposition level. Therefore, E_j represents the wavelet energy for the frequency band ranging from $f/2^{j-1}$ to $f/2^j$.

Fig. 4.6 depicts the wavelet energy of all scales for LDoS attack flow and normal TCP flow. Similar to the PSD distributions for these two types of flows, the wavelet energy of the normal TCP flow is less concentrated than that of LDoS attack flow. Most energy of LDoS attack flows is concentrated in the large scales, which corresponds to the low frequency bands.

Hence, information entropy can also be applied here to identify this difference in wavelet energy distributions. Just like *FPSE* in Section 4.2.1, the wavelet energy needs to be normalized to compute the Wavelet Power Spectrum Entropy (*WPSE*):

$$WPSE = - \sum_j \frac{E_j}{E_{total}} \log \frac{E_j}{E_{total}} \quad (4.12)$$

Figure 4.6: Wavelet Energy for LDoS attacks and normal TCP flows

Thus, for LDoS attack signals, most of the energy will be concentrated within the large scale, which results in a smaller $WPSE$ value compared to that of normal TCP flows. In Section 4.3.2, the author will discuss the $WPSE$ distributions for attack and normal flows.

$FPSE$ and $WPSE$ can both capture the difference in power spectrum distributions for LDoS attacks and normal flows. However, there is some slight difference between these two Power Spectrum Entropies. For the $FPSE$ in Eq. 4.9, the entropy is calculated based on the PSD of discrete frequency points, whereas $WPSE$ in Eq. 4.12 is based on the wavelet power spectrum of frequency bands. Thus, compared to $WPSE$, $FPSE$ should be more sensitive to the LDoS attacks in terms of the different durations and periods of attacks. In Section 4.3, the author will discuss more about the difference between these two metrics based on the evaluation results.

4.2.3 Modification of the two Metrics

Although *FPSE* and *WPSE* can differentiate the LDoS attacks and normal flows, these two metrics may cause some false positives under some special situations in real network. When a flow contains mainly high frequency signals, its power spectrum distributions may have spiky values in the high frequency part. Hence, this flow can also generate a lower value of *PSE*. One example for this kind of flow is the Port Scan. The most common type of Port Scan is TCP half-open scan [32]. It just sends a packet with the SYN flag set and waits for the SYN-ACK from the target, so that it does not need to complete the TCP connection. Owing to this mechanism, TCP half-open scan is a relatively quick scan which can scan thousands of ports per-second. Owing to this high frequency performance, *FPSE* and *WPSE* may generate false alarms when they meet the port scan flows.

In order to address this problem, the author needs to make some modification to the metrics. As can be seen from Fig. 4.4 and Fig. 4.5, both LDoS flows and normal TCP flows contains little energy in the high frequency part. Given this fact, the power spectrum of the high frequency part can just be eliminated when calculating the Power Spectrum Entropy. Therefore, Eq. 4.9 and Eq. 4.12 can be modified into:

$$FPSE = - \sum_{k=1}^{\frac{5}{6}K} p(k) \log p(k) \quad (4.13)$$

$$WPSE = - \sum_{j=2}^J \frac{E_j}{E_{total}} \log \frac{E_j}{E_{total}} \quad (4.14)$$

where K is the maximum frequency of DFT transform and J is the largest wavelet decompose scale. For *FPSE*, the power spectrum of the first $\frac{5}{6}K$ frequencies are taken into account empirically. The author calculates the *WPSE* from E_2 to E_J , as the first

scale mainly contains the high frequency signals.

As there is little energy in the high frequency parts of the LDoS attacks and normal TCP flows, these modifications will only change the values of PSE slightly so that it will not affect the LDoS attack detection performance of PSE . Most of the energy of Port Scan flows is located in the high frequency part, whereas their power spectrum distributions of the low and middle frequency parts are similar to those of the normal TCP flows, which results in similar values in the new PSE . Therefore, by eliminating the high frequency power spectrum, PSE will not mis-classify the Port Scan flows and maintains the same LDoS detection performance.

4.3 Evaluation of Two Detection Metrics

In this section, the author evaluates two detection metrics through simulations in NS-3 simulator [51]. First, the author illustrates the simulation topology in Section 4.3.1. Then, the statistical characteristics of $FPSE$ and $WPSE$ are analyzed in Section 4.3.2, so that the threshold for each metric can be determined.

4.3.1 Simulation Topology

By using NS-3, the author builds a dumbbell network topology to evaluate the performance of the proposed algorithms. As presented in Fig. 4.7, there are two bottleneck routers (Router 1 and Router 2), which are connected with each other by a link with a linkrate of 10 Mbps and a propagation delay of 2 ms. Multiple normal users and attackers are connected to Router 1 with a bandwidth of 100 Mbps. If there are more than 1 attacker in the network ($m > 1$), the attackers can trigger the Low-Rate DDoS (LDDoS) attacks.

Figure 4.7: NS-3 bottleneck network topology

In this topology, normal users will communicate with the server by using TCP-reno packets of 1500-byte in size. UDP protocol is often used by LDoS attackers. Thus, the attacker nodes will generate UDP LDoS traffic with a packet size of 50 bytes. Based on the analysis in [65], the overall transmission rate of attackers can be set to be equal to the bottleneck linkrate, which is 10 Mbps in the network topology. Based on this topology, several simulations will be conducted to evaluate the two detection metrics.

4.3.2 Analysis of Two Information Metrics

In this subsection, the author derives the distributions of Power Spectrum Entropies of the normal TCP flows and LDoS flows so as to determine the detection thresholds for *FPSE* and *WPSE*. For LDoS attacks, the author just sets one attack node to send the LDoS attack packets. In the experiments, the attack rate is set to a constant i.e. $R = 10Mbps$. The author generates two sets of LDoS attack flows by changing the other two parameters (T and D). The first set has the same attack duration $D = 100ms$, while the attack period is varied from $0.6s$ to $2s$. The second set has the same attack period $T = 1.35s$, while their attack duration is varied from $100ms$ to $500ms$. Therefore, the author can calculate the *PSE* of the different types of LDoS

Figure 4.8: *FPSE* Distributions of Normal Flows and Attack Flows

flows.

For the normal TCP flows, the author selects random numbers of normal nodes to communicate with the servers. For each normal node, it sends TCP-reno packets to a random number of servers (maximum 20). The total number of sending packets for each node follows the Gaussian distribution which has a mean of 350000 and standard deviation of 100000. In total, the author evaluates 1000 normal TCP flows.

In Router 1, time series data are extracted for each traffic flow. Here, all flows are defined by a 3-tuple (Source IP, Destination IP, Protocol). The packet arrival number of each flow is sampled at a period of 2ms to obtain the time series data. The time length of each series is 3s. Therefore, based on these time series data, *FPSE* and *WPSE* can be computed for each flow packet. Therefore, the author can derive the distributions of Power Spectrum Entropies of the normal TCP flows and LDoS flows. Based on these distributions, these thresholds can be easily selected for these two detection metrics.

FPSE

Through extensive simulations, the author obtains the distributions of *FPSE* for LDoS attack and normal TCP flows. Fig. 4.8 presents the boxplots of *FPSE* for normal flows and two sets of attack flows. Fig. 4.8(a) shows the *FPSE* comparisons of normal TCP flows and LDoS attack flows with different attack periods. As can be seen from the figures, the smaller the attack periods, the smaller the *FPSE* values of the attack flows. When the attack period decreases, the attack flows display stronger periodicity. This may enlarge the PSD values of the low frequency part, which results in the decrease of the *FPSE* values. Fig. 4.8(b) also shows that the *FPSE* of the attack flows decreases with the increment in attack duration.

These two distributions also offer us empirical guidance on how to select an appropriate *FPSE* threshold. Obviously, a higher *FPSE-Threshold* results in a higher True Positive Rate and a higher False Positive Rate. This higher False Positive Rate can also degrade the overall TCP throughput. Here, Mahalanobis distance is utilized to select an appropriate threshold. The criterion is defined as follows:

$$Dist(x) = D_a(x) + D_n(x) = \sqrt{\frac{(x - \mu_a)^2}{\sigma_a}} + \sqrt{\frac{(x - \mu_n)^2}{\sigma_n}} \quad (4.15)$$

where $D_a(x)$ and $D_n(x)$ are the Mahalanobis distances between the threshold x and the attack and normal *FPSE* distributions respectively. The large value of this criterion ensures that the corresponding threshold can differentiate LDoS attack packets efficiently while generating very few false alarms.

From Fig. 4.8, it can be objectively observed that the *FPSE* distribution of the normal flows is closest to that of attack flows with $D = 100ms$ and $T = 2.0s$. Therefore, by applying Eq. 4.15 to these two distributions, 4.2 is determined as an appropriate *FPSE* threshold to detect the LDoS attacks. Thus, for the following evaluations, the

author selects 4.2 as the *FPSE* threshold. The flow will be determined as a LDoS attack flow when its corresponding *FPSE* is smaller than 4.2.

WPSE

It is also necessary to analyze the distributions of *WPSE* for the attack and normal flows. As *WPSE* takes advantage of wavelet analysis, an appropriate mother wavelet needs to be selected. The best choice of mother wavelet is the one which can separate these two distributions the most. The author defines one criterion here to measure how separable these two distributions are:

$$J = \frac{|\mu_a - \mu_n|^2}{\sigma_a + \sigma_n} \quad (4.16)$$

where μ_a and μ_n are the means of the *PSE* distributions for the attack and normal flows respectively, and σ_a and σ_n are the variances. $|\mu_a - \mu_n|^2$ represents the "distance between two distributions" while $\sigma_a + \sigma_n$ means the "distance within distribution". Therefore, this criterion represents the separation degree of two distributions. The larger this criterion is, the more separable these two distributions are. For the *FPSE* distributions of the normal flows and attack flows with $D = 100ms$ and $T = 2.0s$, J value is 8.3104, which means that there is a marked separation between these two *FPSE* distributions.

Here, the parameters of attack flows are set as: $R = 10Mbps, T = 1.35s, D = 100ms$. Therefore, the *WPSE* distributions of normal TCP flows and attack flows are obtained by using different mother wavelets. Table 4.1 presents the criterion values for different mother wavelets. As seen from the table, *db5* has the largest value of 13.27. Thus, for the following evaluations, *db5* is selected to perform the wavelet analysis.

Fig. 4.9 presents the *WPSE* distributions of the attack and normal flows when

Table 4.1: Distribution Distances for different Mother Wavelets

Mother Wavelet	Separation Degree	Mother Wavelet	Separation Degree
dB2	1.9107	sym2	1.9107
dB3	1.1669	sym3	1.1667
dB4	9.7939	sym4	11.3512
dB5	13.2700	sym5	10.3921
dB6	9.7992	sym6	5.1803
dB7	7.1226	sym7	6.8607
dB8	6,8160	sym8	5.4953
dB9	9.7097	sym9	4.4057
dB10	10.7681	sym10	2.7207

Figure 4.9: *WPSE* Distributions of Normal Flows and Attack Flows

using *db5* mother wavelet. When the attack duration and attack period change, the means of the *WPSE* distributions do not vary a lot. It reveals that the detection performance of *WPSE* seems more stable than that of *FPSE*. As can be seen, the *WPSE* distributions of the LDoS attacks with $D = 300ms$, $T = 1.35s$ is closest to that of the normal TCP flows. Therefore by applying Eq. 4.15 to them, the author obtains 1.1 as the *WPSE-Threshold* to detect LDoS attacks.

4.4 Comparing with Existing LDoS Detection Methods

Although many researchers have proposed many LDoS detection systems, it is difficult for us to conduct a fair comparison among all these different approaches. As the proposed two detection metrics are both based on Power Spectrum Analysis, the author compares the PSE based detection metrics with other power spectrum based detection systems in terms of True Positive Rate (TPR) and False Positive Rate (FPR). Then, the author conducts another comparison with other LDoS detection methods.

4.4.1 Comparing with Power Spectrum based Detection Systems

In [23] [16] and [15], Fourier Power Spectrum Analysis is utilized to detect LDoS attacks. Their systems all rely on *cumulative traffic spectrum (CTS)* to determine the LDoS attack packets. CTS is defined as:

$$\phi(f) = \sum_{i=1}^f PSD(i) / \sum_{i=1}^{\max} PSD(i) \quad (4.17)$$

By comparing this CTS of each incoming flow at the preset pivot frequency (p), their detection systems are able to determine the attack flows. Before the detection, this pivot frequency should be selected by finding the maximum gap point between the CTSs of attack flows and normal TCP flows.

However, this maximum gap point will vary with the change of attack period and duration. If the attacker changes the LDoS attack period or attack duration, this preset pivot frequency cannot guarantee the detection performance. Therefore, their

detection systems are not robust to detect the variations of LDoS attacks.

In contrast, *FPSE* and *WPSE* both combine the power spectrum analysis with the Information Entropy, which takes all the power spectrum values into consideration. Therefore, these two detection metrics do not need to select the pivot frequency beforehand and are thus more robust to the variations of LDoS attacks.

NS-3 Simulation Results

First, the author implements *FPSE*, *WPSE* and *CTS* in NS-3 to conduct the comparisons. For the LDoS attacks which use the attack period $T = 1.35s$ and the attack duration $D = 100ms$, the pivot frequency is selected as $p = 40Hz$ by applying the Algorithm 1 in [23]. *FPSE*, *WPSE* and *CTS* all need to analyze the time signals of network traffic. Thus, the author samples the packet arrival numbers at the router at a sampling period of $2ms$. For all kinds of LDoS attacks, the detection thresholds for *FPSE* and *WPSE* are set as 4.2 and 1.1 respectively.

In this section, the author also applies the network topology in Fig. 4.7. All the three detection metrics are implemented in the Bottleneck Router 1 to classify every incoming packet. For the following evaluations, two normal node sends TCP-reno packets to the server continuously. One attacker sends LDoS attack packets to the server by using UDP protocol.

With two parameters (T, D) for the LDoS attacks, two sets of comparison experiments are conducted by fixing the value of one parameter and changing the other one. For the first set of experiments, the period of LDoS attacks is varied from $0.6s$ to $2s$, while the duration of the attacks is constant at $100ms$. For the other set of experiments, the attack durations are varied from $100ms$ to $500ms$ while the LDoS attack period is set as $1.35s$ constantly. The sending rates of all the LDoS attacks are

Figure 4.10: Detection Results for LDoS Attacks with Different Attack Periods

Figure 4.11: Detection Results for LDoS Attacks with Different Attack Durations

10Mbps. Thus, by changing the attack durations and periods of LDoS attacks, the results of these three methods can be derived, which are presented in Fig. 4.10 and Fig. 4.11.

For the LDoS attacks with different attack periods and durations, two PSE-based detection metrics are quite effective, where the TPRs are all larger than 93% and FPRs are all smaller than 0.5%. The detection thresholds for *FPSE* and *WPSE* are robust to the LDoS attacks with different attack periods and durations. However, the detection performance of CTS-based approach drops significantly with the changes in

attack period and duration. For instance, when the LDoS attack period is $0.6s$, the TPR of CTS-based approach drops to 88.5% and its FPR increases to 2.5%. When the LDoS attack duration is larger than $300ms$, the FPRs of CTS-based approach become more than 2.0%.

These comparison results clearly indicate that PSE-based detection metrics are highly robust to the variations of LDoS attacks compared to CTS-based approach. In the experiments, the pivot frequency ($p_{selected}$) of CTS-based approach is selected for the LDoS attacks whose attack periodicity is $T = 1.35s$ and attack duration is $D = 100ms$. When the attack period decreases or the attack duration increases, the ideal pivot frequency (p_{ideal}) will increase accordingly. Thus, the previous $p_{selected}$ is not suitable for these different LDoS attacks, which results in a severe degradation of its detection performance. Therefore, CTS-based approach is not robust to detect LDoS attacks with different attack periods and durations.

Real Network Trace Detection Results

In addition to conducting NS-3 simulations, the author also evaluates PSE on real traffic traces from the Measurement and Analysis of the WIDE Internet (MAWI) working group to evaluate the detection metrics [45]. From this dataset, the author selects 3 days of data: 20130713, 20140910 and 20150911. From these three days of network trace, any flows with more than 1000 packets are extracted for the evaluation. Therefore, a total of 7210 TCP flows and 4153 UDP flows are extracted. CTS, *FPSE* and *WPSE* are applied to these traffic flows. The classification results are presented in Table 4.2.

When classifying these real normal TCP flows, *CTS* generates 11.827% false alarms. *FPSE* has the False Positive Rate of 5.876%. The False Positive Rate of *WPSE* is only

Table 4.2: Comparisons of False Positive Rate by using MAWI Dataset

	Number	CTS	FPSE	WPSE
TCP Flows	7210	11.827%	5.876%	0.177%
UDP Flows	4153	3.231%	0.740%	0.161%

0.177%. When classifying normal UDP flows, all the methods perform better, among which *WPSE* still has the best performance. In conclusion, *FPSE* and *WPSE* both perform better than *CTS*.

Since there is no real LDoS attack trace which is publicly available, it will be left to the future work to utilize real attack traffic to evaluate the two detection metrics.

Discussions between *FPSE* and *WPSE*

Here, the author has some discussions about *FPSE* and *WPSE*. As analyzed in Section 4.2.2, although these two metrics both can exhibit the periodicity of the signals, there is a slight difference between them. Therefore, this difference results in different detection performance when countering LDoS attacks. With reference to Fig. 4.10 and Fig. 4.11, let us focus on the detection results of *FPSE* and *WPSE*.

As seen from the results, their performances are both better than the CTS-based approach in terms of higher TPRs and lower FPRs. They exhibit similar True Positive Rates when facing LDoS attacks with difference attack periods and durations. However, the False Positive Rates of *WPSE* seems more stable than that of *FPSE*. The evaluation results in Table 4.2 also reveals that *WPSE* is superior to *FPSE*. The advantage of *WPSE* over *FPSE* is mainly because the granularity of the scales in Wavelet Transform is bigger than that of Discrete Fourier Transform. As a result, *WPSE* is able to detect the LDoS attacks with fewer false alarms.

4.4.2 Comparing with other LDoS Detection Systems

In this subsection, the author also compares the detection results of two PSE-based detection metrics with other LDoS detection systems, namely, Chaos Detection [112], Multifractal Detection [115], Adaptive EWMA (AEWMA) based Detection [107], Infinite Impulse Response (IIR) filter based Detection [114], Conditional Random Field (CRF) based Detection [121] and Flow Connection Entropy Series (FCE) Detection [76]. The detection results of these six methods listed in Table 4.3 are quoted from the respective papers.

Chaos Detection takes advantage of the Duffing Oscillator in chaos system to detect the LDoS attacks [112]. By treating LDoS attack and TCP flows as the small signal and the noise signal respectively, Duffing oscillator can be applied here to detect LDoS attacks. The experiments in [112] show that the attacks can be detected in 10s with TPR of 92.00% and FPR of 6.00%. This method can only be applied to detect single LDoS attacks. When it encounters the distributed LDoS attacks, the attack signals will be submerged by the normal flows, in which Duffing oscillator cannot separate the attack signals from the normal flows. Therefore this method cannot detect the distributed LDoS attacks.

Multifractal Detection is actually based on the multifractal detrended fluctuation analysis (MF-DFA) [115]. By utilizing MF-DFA, their detection system is able to exploit and estimate the changes in multifractal characteristics of network flows. This detection method generates 9% False Positive Rate. This is because only the long-lived traffic flows exhibit very strong multifractal characteristic, whereas some short-lived flows do not have this property. Therefore, this Multifractal Detection generates many false alarms.

AEWMA Detection is based on adaptive EWMA algorithm [107]. By utilizing an

adaptive weighting function, AEWMA algorithm can measure the abnormal distribution of network traffic, so as to detect LDoS attacks. However, AEWMA only takes the time domain into consideration. Therefore, it produces 7.27% false positives.

In IIR Filter based Detection [114], Wu et al. attempt to utilize IIR filter to detect LDoS attacks. By transforming the time series from the time domain into the frequency domain, IIR filter is able to filter out the LDoS attacks. However, this method has a very low detection accuracy and it also generates many false alarms.

In CRF based Detection [121], Yi et al. adopted the conditional random field to detect LDoS attacks. This method chooses classification features of LDoS attack firstly. After being trained on a large number of sample traffic, this CRF model can be utilized to detect the LDoS attacks. This method achieves the highest TPR (95.83%). However, it produces more than 4% false positives. In addition, the time complexity of CRF's training process is very large ($O(n^3)$).

In FCE Detection [76], the author decomposes the FCE data into the trending and steady random components. By analyzing the long-term and short-term trends in the traffic, their system is able to detect the LDoS attacks. TPR and FPR of this system are 90.20% and 4.30% respectively. As this method requires the analysis of the long term trends of the network traffic, its average detection latency is much longer than the other detection systems.

For *FPSE* and *WPSE*, their TPRs and FPRs are presented when detecting LDoS attacks with $T = 1.35s$ and $D = 100ms$. As seen from Table 4.3, *FPSE* and *WPSE* both achieve very high TPR (more than 95%) and low FPR (smaller than 0.5%). As the sampling time of these two metrics is 3 seconds, PSE-based detection metrics are supposed to detect the attacks in less than 3s. Owing to the use of Fourier Transform in *FPSE*, its time complexity is $O(n \log n)$, whereas the time complexity of *WPSE* is

$O(n)$ due to the wavelet analysis.

Comparing the two Power Spectrum Entropy metrics with other methods, NS-3 simulation results and real trace evaluation results show that both *FPSE* and *WPSE* have better detection performances in terms of higher TPRs and lower FPRs. On top of that, *WPSE* is more robust to detect the different LDoS attacks and has a smaller time complexity, which makes it a better LDoS detection metric.

Table 4.3: Comparison of Detection Results

Detection System	TPR	FPR	Detection Latency(s)	Time Complexity
Chaos Detection	92.00%	6.00%	10	$O(n)$
Multifractal Detection	92.00%	9.00%	unknown	$O(n \log n)$
AEWMA Detection	unknown	7.27%	20	$O(1)$
IIR Filter Detection	81.36%	7.45%	Unknown	$O(n \log n)$
CRF Detection	95.83%	4.19%	5	$O(n^3)$
FCE Detection	90.20%	4.30%	32	$O(n)$
FPSE	95.27%	0.35%	< 3	$O(n \log n)$
WPSE	95.32%	0.18%	< 3	$O(n)$

4.5 PSE-based Robust RED (PRRED) Algorithm and its evaluations

The evaluation results in the previous section show that these two detection metrics can be utilized to detect LDoS attack packets efficiently with very few false alarms. In this section, based on these two metrics, the author proposes a novel LDoS mitigation algorithm which combines PSE and Robust Random Early Detection algorithm [125]. Then, the author evaluates the mitigation results based on the simulations in NS-3. Section 4.5.2 illustrates the NS-3 simulation settings of different queuing algorithms. The evaluations of PSE-based Robust RED algorithms are based on two network scenarios: transmitting long-lived and short-lived TCP flows in Section 4.5.3 and Section

4.5.4 respectively.

4.5.1 Architecture of PRRED LDoS Mitigation Algorithm

In this subsection, the author illustrates the architecture of a novel LDoS mitigation algorithm which combines Power Spectrum Entropy and Robust-RED algorithm to detect and mitigate LDoS attacks. Robust-RED (RRED) algorithm is first introduced in [125]. This RRED algorithm determines if one packet is an attack packet or not based on its arrival time. If this packet arrives at the router within a short period after some previous packets are dropped, this packet will be considered as an attack packet and will be dropped. This is just a rough detection of LDoS attacks. LDoS attack and normal TCP packets may arrive at the bottleneck router simultaneously. Therefore, RRED's rough detection may result in a high false positive rate which will force more normal TCP packets to be dropped. As a consequence, these additional TCP packet drops may degrade the throughput of normal users.

Fig. 4.12 presents the basic architecture of the proposed PSE-based Robust-RED (PRRED) queuing algorithm.

At the start, similar to RRED, the algorithm assigns a score to each traffic flow to roughly determine if this flow is a suspicious attack flow or not. To be the same with Section 4.3, the 3-tuple is still utilized to define a flow. Let T_{drop} be the time of the last packet dropped by PRRED, and Δt represents the short time period. Thus, any packet that arrives at the bottleneck router within the time range $[T_{drop}, T_{drop} + \Delta t]$ will be considered as a suspicious attack packet. Then, the score of the corresponding flow will be decreased by one. Otherwise, this score will be increased by one. Any packet which comes from the flow with a non-negative score will be delivered to the RED block. When the score of one flow achieves a negative value, this flow will be considered as

Algorithm 1 Pseudo code of PRRED algorithm.

```

PRRED-ENQUEUE(pkt)
   $T_{drop} \leftarrow \text{MAX}(f(pkt).T_1, T_2)$ 
  if  $t_{arrival} \in [T_{drop}, T_{drop} + \Delta t]$  then
     $f(pkt).score--$ 
  else
     $f(pkt).score++$ 
  end if
  if  $f(pkt).score \leq 0$  then
    calculate  $f(pkt).Entropy$ 
    if  $f(pkt).PSE \leq \text{Threshold}$  then
      drop(pkt)
       $f(pkt).T_1 \leftarrow t_{arrival}$ 
    else
      if !RED-ENQUEUE(pkt) then
         $T_2 \leftarrow t_{arrival}$ 
      end if
    end if
  else
    if !RED-ENQUEUE(pkt) then
       $T_2 \leftarrow t_{arrival}$ 
    end if
  end if
return

```

Figure 4.12: Architecture of PSE-based Robust-RED

a suspicious attack flow. Any packet from this flow will be fed to the *PSE*-based Filtering block. Based on *FPSE* or *WPSE*, this *PSE*-based Filtering block will further determine whether this is an attack flow or not. By comparing this flow's *PSE* with the preset threshold, the flow can be easily and accurately determined whether it is an attack flow.

In summary, the initial Arrival Time based Detection described above can identify a suspicious subset of packets for PRRED's later processing. This is to avoid computing *PSE* for each incoming packet thereby reducing the time complexity of the algorithm. In addition, *PSE* Filtering block will in turn help increase the detection accuracy.

Algorithm 1 presents the pseudo code of the proposed PRRED algorithm. Based on the detection metric chosen, there are two kinds of PRRED algorithms: *FPSE*-based Robust RED (FRRED) and *WPSE*-based Robust RED (WRRED).

Table 4.4: Parameter Settings for Different Queuing Algorithms

Algorithms	Queue Size (pkts)	Δt (ms)	t_{sample} (ms)	Sampling Time (s)	FPSE Threshold	WPSE Threshold	Mother Wavelet
RED	100	NA	NA	NA	NA	NA	NA
RRED	100	10	NA	NA	NA	NA	NA
FRRED	100	10	2	3	4.2	NA	NA
WRRED	100	10	2	3	NA	1.1	db5

4.5.2 Simulation Settings of Different Queuing Algorithms

Based on the topology in Section 4.3.1, several simulations are performed to evaluate the performance of different queuing algorithms when countering LDoS attacks. In the experiments, two normal users communicate with the server by sending TCP-reno packets. The attackers will send UDP LDoS attacks to the server. The aggregate attack rate (R) is set at the same value as the bottleneck linkrate.

Four kinds of RED queuing algorithms are implemented in the bottleneck Router 1: RED [44], RRED [125], FRRED and WRRED. The parameters for these four algorithms are listed in Table 4.4. The queue sizes of all the queuing algorithms are set to 100 packets. For RRED, FRRED and WRRED, Δt is set as 10 ms which is the same as in [125] for comparison. For FRRED and WRRED, time series data need to be collected to analyze their frequency characteristic. Thus, the packet arrival numbers are sampled at a period of 2ms ($T_{sample} = 2\text{ms}$). As per the analysis in Section 4.3.2, FPSE threshold and WPSE threshold are chosen as 4.2 and 1.1 respectively. In addition, *db5* is selected to perform the wavelet analysis in WRRED. It should be noted that the author will set the same parameters for all the following evaluations.

4.5.3 Long-Lived TCP Traffics

In this subsection, the mitigation performances of different queuing algorithms are evaluated based on the normalized goodput. The normalized goodput is defined as the ratio of the goodput achieved by the TCP flow under the LDoS attacks and the TCP goodput without the attacks. Each simulation lasts for 300s. The normal users continuously send packets to the server, whereas the attackers send LDoS attacks from 5s.

Single LDoS Attack

The author uses only one attack node to send LDoS attack packets to the bottleneck router. Its transmission rate is set as 10 Mbps, which is equal to the bottleneck linkrate. The author will also conduct two sets of experiments by fixing one of the two parameters (Attack Period T and Duration D) and changing the other one. The comparison results of the two sets of experiments are presented in Fig. 4.13. Fig. 4.13(a) depicts the goodput comparisons of different queuing algorithms when countering LDoS attacks with different attack periods.

Apparently, RED achieves the worst goodput. When the attack period increases, there are two "null points" for the normalized goodput of RED, which is consistent with the analysis of Eq. 4.2 in [65]. These two "null points" happen when the LDoS attack periods $T = RTO_{min}$ and $T = RTO_{min}/2$. Therefore, RTO_{min} of the simulation network is approximately 1.35s. RRED does not perform very well when facing LDoS attacks with different attack periods. When the attack period is very small, the attacker sends packets to the router more frequently which results in more frequent congestions. RRED may drop more TCP packets as their arriving time is quite close to that of the attack packets. Thus, the normalized goodput of RRED is quite low when the attack

Figure 4.13: Goodput of Different Mitigation Algorithms under variations of LDoS attacks

period is small. For the LDoS with large attack period, RRED can protect nearly 90% of the bottleneck bandwidth. As for FRRED and WRRED, by adding the *PSE*-based Filtering module, these two algorithms can fully distinguish LDoS attack packets from the normal TCP packets. Thus, less normal packets will be dropped compared to RRED. Therefore, FRRED and WRRED can protect almost all the bottleneck bandwidth.

Fig. 4.13(b) depicts the performance of different queuing algorithms when counter-ing LDoS with different attack durations. FRRED and WRRED are still more effective than the other two algorithms. They both protect nearly the whole bottleneck bandwidth.

Low-Rate Distributed DoS (LDDoS) Attacks

Here, 20 attack nodes are set to send LDoS attack packets so as to trigger Low-Rate Distributed DoS (LDDoS) attacks. The sending rate of each attack node is set to 0.5 Mbps, so that the aggregate attack rate is equal to the bottleneck linkrate of 10 Mbps.

Figure 4.14: Goodput of Different Mitigation Algorithms under variations of LDDoS attacks

The author also conducts two sets of experiments (different attack periods and different attack durations) to evaluate the performance of the algorithms. The evaluation results are presented in Fig. 4.14. When countering LDDoS attacks with different attack periods and durations, FRRED and WRRED both have a better mitigation performance compared to RRED. Due to the smaller attack rate of each attack flow, *FPSE* generates more false alarms compared to *WPSE*. Therefore, WRRED can better protect the normal TCP bandwidth than FRRED when countering LDDoS attacks.

4.5.4 Short-Lived TCP Traffic

As of now, the author has considered the scenario with long-lived TCP flows and LDoS attack flows. Here the author will study the situation when transmitting the short-lived TCP flows. The topology in Section 4.3.1 is still adopted here. One attack node will send LDoS attack packets to the bottleneck router. The parameters of LDoS attacks are set as: $T = 1.35s$, $D = 0.1s$, $R = 10Mbps$. The normal user will send files with different file sizes. For each size, the file will be sent 10 times with random

Figure 4.15: Evaluation Results for Transmission of Short-lived TCP Traffic

starting time. Thus, the transmission time can be measured and averaged for the files of different sizes.

Firstly the transmission time for different files is calculated when there is no attack in the system. For this situation, RED queue is utilized in the bottleneck router. These transmission time are set as the baseline to evaluate other queuing algorithms under attacks. The algorithms are tested with two different RTT for normal TCP senders i.e., 20ms and 60ms. The evaluation results are presented in Fig. 4.15. For the RED algorithm, it will take much longer time to transmit the file than the other three algorithms when countering LDoS attacks. For RRED, it will take almost double the time of the baseline to transmit the file whose size is larger than 1000 pkts. FRRED and WRRED both take less time to transmit files than RED and RRED. For WRRED, it takes the least transmission time for both experiments.

4.5.5 Conclusion of PSE-based Robust RED algorithms

In conclusion, FRRED and WRRED both can detect and mitigate LDoS attacks swiftly and efficiently. The better robustness of *WPSE* does help WRRED to achieve better

mitigation results especially when countering LDDoS attacks. In addition, the computational complexity of FRRED is $O(n \log n)$ due to the Fourier Analysis, whereas the computational complexity of WRRED is $O(n)$ due to the Wavelet Analysis. WRRED can thus mitigate LDoS attacks faster and more efficiently than FRRED in real time. Therefore the author recommends to apply WRRED in the bottleneck routers to detect and mitigate LDoS attacks.

4.6 Conclusion

In this chapter, the author introduces two novel information metrics to detect LDoS attacks: Fourier Power Spectrum Entropy and Wavelet Power Spectrum Entropy. These two metrics can detect the Low-Rate DoS attacks efficiently. Moreover, the author proposes a LDoS attack mitigation algorithm: Power Spectrum Entropy based Robust RED algorithm.

By evaluating on NS-3 simulations and real network traces, the detection results show that the two metrics are very promising in detecting the LDoS attacks with very few false positives, especially *WPSE*. Furthermore, the proposed mitigation algorithms can effectively protect the normal TCP bandwidth. In conclusion, the proposed information metrics and LDoS mitigation algorithms can substantially detect and mitigate the Low-Rate DoS attacks. As WRRED has a low computational complexity and a better performance when countering LDDoS attacks, it is better to apply the WRRED algorithm in the bottleneck routers.

Chapter 5

Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection

This chapter presents an image outlier detection algorithm which combines autoencoder with Adaboost (ADAE). By repeatedly training on the weighted normal dataset, ADAE is able to obtain a sequence of weak autoencoders. The final prediction model of ADAE is a weighted summation of all the weak autoencoders. As each successive autoencoder attempts to capture the correlations that the previous one has failed to capture, ADAE can better capture the image feature correlations than the single autoencoder. In addition, the author introduces the Sparse Group Lasso (SGL) constraint to the learning objective of the autoencoder in order to eliminate unnecessary parameters.

The main contributions of this chapter are threefold. Firstly, the author proposes a novel image outlier detection model which combines autoencoder with Adaboost. By evaluating on several popular image datasets, this model has been shown to have better detection performance than the other detection methods. Secondly, the author introduces the Sparse Group Lasso constraint into ADAE (ADAE-SGL) to obtain a compact detection model. In order to optimize this additional objective of ADAE-SGL, the author combines Adagrad with Proximal Gradient Descent (Ada-PGD). Thirdly, in order to determine the best penalty factors of SGL, the author reformulates the objective of ADAE-SGL into a bi-objective optimization problem. Therefore, by utilizing evolutionary multi-objective optimization method, the author successfully finds the best penalty factors. The evaluation results show that ADAE-SGL achieves similar detection performance while obtaining a compact detection model.

5.1 Adaboost-Autoencoder

In this section, the author details the Autoencoder-based outlier detection algorithm first. Then, the author proposes an extended image outlier detection algorithm which combines Adaboost with autoencoder.

5.1.1 General Autoencoder-based Outlier Detection

Autoencoder is a simple neural network which only has one hidden layer. Normally, it consists of two parts: the *encoder* and *decoder*

Encoder : Assume that the training set is $\{x_1, x_2, \dots, x_n\}$, which are all d dimensional vectors ($x_i \in R^d$). The *Encoder* is trying to encode the original data into m ($m < d$) number of neurons which make up the hidden layer. The activation output of the

hidden neuron i is:

$$a_i = f_\theta(x) = s\left(\sum_{j=1}^n W_{ij}^{input} x_j + b_i^{input}\right) \quad (5.1)$$

where x is the input vector, s is a non-linear activation function, θ is the parameters $\{W^{input}, b^{input}\}$, W represents the encoder weight matrix with size $m \times d$ and b is a bias vector of dimensionality m . Therefore, the input vector is encoded into a lower-dimensional vector.

Decoder : The resulting hidden representation a is then decoded back to the original input space R^d . The mapping function is as follows:

$$x'_i = h_{\theta'}(a) = s\left(\sum_{j=1}^n W_{ij}^{hidden} a_j + b_i^{hidden}\right) \quad (5.2)$$

The parameter set of the decoder is $\theta' = \{W^{hidden}, b^{hidden}\}$.

The autoencoder is optimized to minimize the average reconstruction error with respect to θ and θ' :

$$\theta^*, \theta'^* = \arg \min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^n \varepsilon(x_i, x'_i) \quad (5.3)$$

where ε is the square error function.

As stated in [95], outlier detection using autoencoder is based on the assumption that there are strong correlations between the features of the normal data. In the training phase, the training dataset consists of all normal data. Hence, by minimizing the reconstruction errors, the autoencoder is able to find a subspace which can capture the statistical correlations of the normal data. In the test phase, the test data are fed into this autoencoder and their reconstruction errors will be calculated. As the outliers are images from different semantic concepts, they are less likely to be reconstructed well, resulting in relatively larger reconstruction errors than the normal data.

As a result, by thresholding these reconstruction errors, the algorithm can easily

identify the outlier image data:

$$c(x_i) = \begin{cases} normal & \varepsilon_i < \theta \\ outlier & \varepsilon_i > \theta \end{cases} \quad (5.4)$$

5.1.2 Combining Autoencoder with Adaboost

When applying the general autoencoder method to the image outlier detection, the detection results do not meet the expectation. This is because one autoencoder is inadequate to capture all the statistical correlations of the normal data, especially in high-dimensional image dataset. In this subsection, the author proposes a novel image outlier detection algorithm which combines Adaboost with autoencoder. This Adaboost-Autoencoder (ADAE) outlier detection algorithm is presented in Algorithm 2.

Algorithm 2 Adaboost-Autoencoder (ADAE) Outlier Detection

Input: Dataset X of size n , the maximum number of iterations N , the initial weight $\vec{w}(w_i = 1/n)$

for $t = 1, \dots, N$ **do**

1. Train t -th autoencoder (h_t) with the training set X and current weights $\vec{w}$: $\arg \min_h \frac{1}{n} \sum_i w_i \varepsilon(x_i, h(x_i))$
2. Calculate the adjust error e_i^t for each datum: Let $D_t = \max_{i=1}^n \varepsilon(x_i, h_t(x_i)) = \max_{i=1}^n \sum_j (x_{ij} - h_t(x_{ij}))^2$, then $e_i^t = \frac{\varepsilon(x_i, h_t(x_i))}{D_t}$
3. Calculate the average error of h_t : $e_t = \sum_{i=1}^n w_i^t e_i^t$
4. Calculate the error rate of h_t : $\beta_t = e_t / (1 - e_t)$
5. Update the weight vector $w_i^{t+1} = w_i^t \beta_t^{1-e_i^t} / Z_t$

end for

Output: $h_f(x) = \sum_{t=1}^N (\ln \frac{1}{\beta_t}) h_t(x)$

The outlier score is $\varepsilon(x, h_f(x))$

At the very beginning, initial weights $w_i = 1/n$ are assigned to each training dataset, so that the first autoencoder is trained on the data in the usual manner. For each successive iteration, the weight for each training data is modified according to its corresponding reconstruction error. Those data that have larger reconstruction error in the previous step have their weights increased, whereas the weights are decreased for those with smaller reconstruction error. Each successive autoencoder is thereby forced to concentrate on those data that have not been reconstructed well by the previous ones. Thus, during the training process of ADAE, the autoencoders attempt to capture the correlations that the previous ones fail to capture. Finally, all of these autoencoders are combined through a weighted summation. The weight for each autoencoder represents the contribution of each respective autoencoder to the final model.

By assembling all the autoencoders, the proposed algorithm can better capture the statistical correlations among the features of the normal data. Therefore, ADAE can achieve much better outlier detection accuracy compared to the general Autoencoder-based method.

5.2 Multi-objective Optimization of Regularization with Sparse Group Lasso

As ADAE utilizes a sequence of autoencoders to capture the statistical correlations among the features of the normal image data, ADAE should have an excellent image outlier detection performance. However, it has a drawback. As it consists of many autoencoders, there are too many parameters in the model, which may result in large computational complexity for detecting image outliers. Therefore, in this section, the author introduces the group sparsity regularization [96] during the training process to

make the detection model compact.

By adding the regularization term, the training objective of each autoencoder becomes:

$$\min_w L(w) = \min_w \varepsilon(\{w\}, D) + \lambda \Omega(w) \quad (5.5)$$

where D is the training dataset, ε is the reconstruction loss parameterized by weights w , Ω is the pre-defined regularization term, and λ is the penalty term which balances the effects of loss function and regularization term.

5.2.1 Sparse Group Lasso

The most famous regularization term is l_2 regularization [127], which is also denoted as "weight decay": $\Omega(w) \triangleq \|w\|_2^2$. Although l_2 regularization term does reduce the magnitude of weights, it cannot generate a sparse solution. The only way to obtain sparsity with weight decay is to set the weights to zero whose amounts are smaller than a pre-defined threshold. However, the threshold is not only difficult to choose, but also not suitable for the real application.

The second most common method is l_1 -norm regularization term [106], which is to penalize the absolute value of the weights:

$$\Omega(w) \triangleq \|w\|_1 = \sum_k |w_k| \quad (5.6)$$

Apparently, l_1 norm is not differentiable at 0. Therefore, it is necessary to apply sub-gradient decent or proximal gradient descent [37] to optimize the objective function. As presented in [37], the proximal map of l_1 norm is the soft-thresholding operator. Thus, the update of this proximal gradient can lead to sparse solutions.

In order to make the neural network compact, it is necessary to prune the unnec-

essary neurons whose incoming or outgoing weights are all 0. In this sense, l_1 norm is not suitable to obtain a compact neural network. Here, the author adopts the Sparse Group Lasso (SGL) regularization term which is introduced in [96]. SGL consists of two parts: l_1 norm and group sparsity factor:

$$\Omega_{SGL}(w) \triangleq \mu\Omega_{GS}(w) + (1 - \mu)\Omega_{l_1}(w) \quad (5.7)$$

where μ is the hyper-parameter for balancing the effects of l_1 norm and group sparsity term. Group sparsity term is defined as the summation of group l_2 norms:

$$\Omega_{GS}(w) \triangleq \sum_g \|W_g\|_2 = \sum_g \sqrt{\sum_i w_{g,i}^2} \quad (5.8)$$

where W_g is the weight matrix for group g . In the implementation, the author puts the outgoing weights from the same neuron into one group. Since l_2 norm has the effect that leads to similar value of weights in one group, minimizing this group sparsity term will result in complete zero of some groups' weights, so that such surplus neurons can be removed. Therefore, by adding this group sparsity regularization term, the optimizer can decide which neuron is effective at each layer.

By adding l_1 norm, SGL can also guarantee the sparsity of weights among the remaining neurons after removing the surplus neurons. The illustrations of sparsity of l_1 norm, group sparsity and SGL are presented in Fig. 5.1. As can be seen from the figure, SGL can remove some neurons completely. At the same time, it can result in a sparse solution for the other active neurons.

By introducing SGL, the training objective in Eq. 5.5 becomes:

$$\min_w L(w) = \min_w \varepsilon(\{w\}, D) + \lambda[\mu\Omega_{l_1}(w) + (1 - \mu)\Omega_{l_{GS}}(w)] \quad (5.9)$$

Figure 5.1: **Illustration of Different Sparse Regularizers** The dash lines represent the connection with zero weights. The white circles denote the removed neurons.

Therefore, by utilizing this additional objective to train each autoencoder in ADAE, it will obtain a compact detection model, which can be referred to as ADAE-SGL.

5.2.2 Adagra Proximal Gradient Descent (Ada-PGD)

As SGL contains l_1 and l_2 norms, SGL is not differentiable at some points. Therefore the proximal gradient descent is utilized to optimize the regularized training objective. Proximal gradient descent consists of two steps. Firstly, it computes an intermediate solution $w_{t+\frac{1}{2}}$ by performing gradient descent to the reconstruction loss error $\varepsilon(\{w\}, D)$. The second step is to compute the proximal map of SGL:

$$w_{t+1} = \arg \min \left\{ \frac{1}{2} \|w - w_{t+\frac{1}{2}}\|^2 + \lambda \eta_t \Omega_{SGL}(w) \right\} \quad (5.10)$$

where η_t is the t -th learning rate and λ is the penalty factor of regularization term.

Based on the work in [37], the proximal mapping for the group sparsity is obtained

as follows:

$$PG_{GS}(W, \eta) = (1 - \frac{\lambda\eta}{\|W_g\|_2})_+ W_g \quad (5.11)$$

where $\|W_g\|_2$ is the l_2 norm of the weights of group g . From this proximal mapping, one can find out why group sparsity has the effect of completely removing some neurons. During this proximal update, whenever the l_2 norm of some group is smaller than $\eta_{t+\frac{1}{2}}$, the weights in this group will be all set to zero, which leads to the removal of the entire group.

The proximal mapping for the l_1 norm is given as follows:

$$PG_{l_1}(W, \eta) = \text{sign}(W)[|W| - \lambda\eta]_+ \quad (5.12)$$

By combining Adagrad with this proximal gradient, the author proposes Adagrad Proximal Gradient Descent (Ada-PGD) algorithm to optimize the training objective. Adagrad is an adaptive learning rate optimization method proposed by [36]. The traditional Gradient Descent takes the learning rate constantly for all parameters, but Adagrad adjusts the learning rate for each individual parameter. For each parameter, its current learning rate is divided by the l_2 -norm of its previous gradients. Thus, if the l_2 -norm is large, the learning rate will be small. Otherwise, the learning rate will be large. Therefore, Adagrad can adaptively control the learning rate of each parameter. The detailed Ada-PGD algorithm is presented in Algorithm 3.

5.2.3 Evolutionary Multi-objective Optimization of the Training Objective with SGL

SGL contains two hyper-parameters λ and μ , which need to be determined in advance. Inspired by the work in [59], the author adopts the Pareto-based learning approach to

Algorithm 3 Adagrad Proximal Gradient Descent Algorithm for Optimizing SGL Regularization

Input: Dataset D , general learning rate η

while $t = 1, \dots, N$ **do**

for each mini-batch data **do**

1. Get the learning rate for the current step: $\eta_{curr} = \frac{\eta}{\sqrt{cache + \Delta}}$. The variable $cache$ keeps track of the square summation of each parameter's gradient and Δ is a smoothing term to avoid division by zero.
2. Update the parameters with gradient descent: $W_{t+\frac{1}{2}} = W_t - \eta_{curr} \nabla \varepsilon(W)$
3. Update the parameters with Eq.5.11: $W_{t+\frac{1}{2},GS} = PG_{GS}(W_{t+\frac{1}{2}}, \eta_{curr})$
4. Update the parameters with Eq.5.12: $W_{t+\frac{1}{2}} = PG_{l_1}(W_{t+\frac{1}{2},GS}, \eta_{curr})$

end for

end while

select the best λ and μ .

The training objective in Eq. 5.9 can be reformulated into a Pareto-based multi-objective optimization problem:

$$\min_w L(w) = \min_w (\varepsilon(w), \Omega_{SGL}(w)) \quad (5.13)$$

Therefore, Pareto-based learning framework can be utilized here to obtain the best sparse constraint penalty factors (λ and μ).

The detailed learning framework is presented in Fig. 5.2. In attempting to find the best penalty factors, the mutation operations are only applied to vary λ and μ in this framework. After mutation, Ada-PGD is employed to optimize the training objective Eq. 5.9. According to the evaluations of $\varepsilon(w)$ and $\Omega_{SGL}(w)$, the population of the next generation can be selected. The selection method from Non-dominated Sorting Genetic Algorithm II (NSGA-II) [33] is utilized here. It mainly contains four steps. First, the offspring is generated by using simulated binary crossover and polynomial mutation. Parents are selected using a binary tournament selection leveraging between

Figure 5.2: Framework for Multi-objective Optimization of SGL Using NSGA-II

the non-dominated sorting and crowding distance. Combine the parent and offspring populations.. Second, the algorithm performs non-dominated sorting and calculate the crowding distance for the combined population. Third, the population is sorted according to the non-dominated sorting and the crowding distances. Fourth, NSGA-II selects the best half of the combined population as the parent of the next generation.

Finally, this Evolutionary Multi-Objective (EMO) based Optimization framework is able to achieve an amount of Pareto-optimal solutions that make up the Pareto Front. The knee point solution of the Pareto Front is selected as the final solution to determine the optimal λ and μ .

5.2.4 Conclusion of the Detection Model

Fig. 5.3 presents the whole diagram of the ADAE detection model. The EMO-based optimization framework can help to obtain the best penalty factors of SGL (λ_{opt} and

μ_{opt}). Therefore, by feeding the deep learning features of images, Ada-PGD can be utilized to optimize the ADAE with SGL constraint. Finally, a compact image outlier detection model can be obtained.

Figure 5.3: The Diagram of ADAE with Sparse Group Lasso Constraint

5.3 Evaluations

5.3.1 Image Datasets and Evaluation Metrics

The following image datasets are utilized to evaluate ADAE.

- **Caltech-101** [42] is an image dataset which consists of images of objects belonging to 101 categories. In the experiments, 10 categories of image data are used to evaluate the proposed algorithm. The size of each image is roughly 300×200 pixels.
- **Scene_data** [70] consists of fifteen natural scene categories. All the categories are included here for the evaluation, and each category contains about 300 image data on average.
- **STL-10** [29] is an image recognition dataset which contains 10 classes of image data. All the image data are utilized in the evaluation.

Table 5.1: Detail Information about all the Evaluation Datasets

	Trainset	Testset	
	Average number of Normal Images	Average number of Normal Images	Average number of Outlier Images
Caltech-101	500	300	460
Scene_data	200	100	500
STL-10	500	900	900
ImageNet	800	520	495

- **ImageNet Data** [34] is a large visual database designed for object recognition research. 10 classes of image data from the ImageNet website are extracted for the evaluation.

In the evaluation, the normal data are all selected from one category of images, whereas the outliers are the images of other categories. The training datasets contain only the normal data. The test dataset consists of both normal and outlier data. The detail information of all the evaluation datasets is listed in Table 5.1. VGG [102] is utilized here to extract the deep learning features of each image in each dataset. In the evaluations, the 4096-dimensional outputs of the first fully-connected layer in VGG are utilized as the input of the proposed ADAE algorithm.

In the following evaluations, two evaluation metrics are mainly adopted: AUC score and F1 score. AUC score is the area under the ROC curve. With different values of thresholds, it is able to obtain the ROC curve so as to get the AUC score. In addition, for all of these thresholds, their corresponding F1 scores can also be calculated. Among these F1 scores, the best F1 score is selected for evaluation.

5.3.2 Methods to be compared

The author also implements several well-known outlier detection methods to compare with the proposed algorithm:

- **Autoencoder (AE)** As introduced in Sec. 5.1.1, the reconstruction errors are treated as outlier scores. For AE and every autoencoder in ADAE, the author only sets 20 hidden neurons and utilizes the ReLU activation function. The detection result of AE is presented here as a baseline result.
- **Local Outlier Factor (LOF)** [13] By measuring the local density of a data point with respect to its neighbors, the local outlier factor (LOF) is introduced to detect anomalies. This LOF is a local factor which only takes into account a restricted neighborhood of each data point. Therefore, LOF scores can be utilized as the outlier scores to classify the outlier data. LOF is a classic outlier detection algorithm. Therefore it is selected here to compare with the proposed ADAE algorithm.
- **Isolation Forest (IF)** [75] As the outliers are few and different from the normal data, they are more susceptible to be isolated. Hence, the outliers can be isolated much closer to the root of IF. Therefore, the number of splittings required to isolate the image data is taken as the outlier score. IF is implemented based on Random Forest, which is another well known type of ensemble learning method, with respect to Adaboost.
- **One-class SVM (OCSVM)** [98] Given a set of normal data, OCSVM attempts to find a soft boundary of that set. Therefore this boundary will help to classify the new point based on whether the point belongs to that set or not. In the evaluation, the distance to the boundary is taken as the outlier score. The points inside the boundary will be assigned a negative distance, whereas the points outside the boundary will have a positive distance. OCSVM is another classic algorithm with very high accuracy.
- **Least Square Approach (LS)** [91] Based on a squared-loss function, a probabilistic

and nonparametric method is proposed for outlier detection. It is a kernel-based method which is similar in essence to OCSVM. As the proposed objective function has a simple analytical solution, it is faster to train than OCSVM. For this method, the outlier probability is adopted as the outlier scores. As a variation of OCSVM, LS can achieve better detection accuracy with less running time. It is the latest outlier detection method, so that it is implemented here to compare with ADAE algorithm.

For all these methods, their corresponding outlier scores can be obtained. Based on these outlier scores, the corresponding AUC and F1 scores can be calculated. Therefore, the author compares their detection performance in terms of AUC and F1 scores.

Figure 5.4: AUC scores and Coefficients for different number of autoencoders

5.3.3 Number of Iterations in ADAE

There is one hyper-parameter in ADAE that needs to be decided in advance: the number of iterations (N) in ADAE. Although more iterations can help ADAE increase the detection accuracy, it also increases the number of autoencoders which results in parameter redundancy of the detection model. Therefore, more iterations will result in greater computational costs in both training and testing.

Here the image data from 'goshawk' category in ImageNet is adopted as the example dataset. By evaluating ADAE algorithm on this example dataset, one can calculate the AUC scores for ADAE with different iteration numbers, which are presented in Fig. 5.4. As the iterations of ADAE progresses, the AUC score steadily increases, reaching 0.996 after 20 iterations. 20 iterations are enough for ADAE to achieve good detection performance. In addition, Fig. 5.4 also shows the coefficient of each autoencoder. As the number of iterations increases, the coefficient keeps on decreasing. The coefficient for the 20th autoencoder is nearly 0, which means that the autoencoders after the 20th iteration have very little influence on the final detection model. Therefore, adding more iterations for ADAE only improves the detection accuracy slightly.

In conclusion, for the following evaluations, ADAE will be just trained with 20 iterations.

5.3.4 Evaluations of ADAE

In this subsection, the author compares the detection performances of the different outlier detection algorithms by evaluating on the image datasets. Comparison results on each category of STL-10 dataset are presented in Fig. 5.6 and Fig. 5.7. The average AUC and F1 scores of the different methods are listed in Table 5.2. From these comparisons, the following observations can be realized:

Table 5.2: Average AUC and F1 scores on STL-10 Dataset

Methods	ADAE	AE	OCSVM	IF	LS	LOF
AUC Scores	0.9743	0.7411	0.9168	0.9096	0.8874	0.70833
F1 Scores	0.9299	0.7578	0.8555	0.8561	0.8470	0.7181

Table 5.3: Detection Results of Different Algorithms on Caltech-101, Scene_data and ImageNet. Evaluation metrics are averaged over all categories

Dataset	Metrics	ADAE	AE	OCSVM	IF	LS	LOF
Caltech-101	Precision	1.0000	0.9703	0.9515	0.9957	0.9808	0.8740
	Recall	1.0000	0.9957	0.9804	0.9978	0.9978	0.9196
	F1	1.0000	0.9828	0.9657	0.9967	0.9892	0.8962
Scene_data	Precision	0.9722	0.8863	0.9081	0.9257	0.8853	0.6819
	Recall	0.9780	0.9820	0.9820	0.9720	0.9880	0.9560
	F1	0.9751	0.9317	0.9371	0.9483	0.9338	0.7960
ImageNet	Precision	0.8709	0.6036	0.7679	0.7786	0.7223	0.5060
	Recall	0.9401	0.9244	0.8857	0.8962	0.8646	0.8610
	F1	0.9050	0.7186	0.8200	0.8314	0.7819	0.6374

- Among all these 6 detection methods, LOF has the worst detection performance. This is because LOF needs to calculate the local density of k nearest neighbors, which is strongly influenced by the curse of dimensionality [62]. In the experiments, the author utilizes 4096-dimensional deep learning features to represent each image. Therefore, LOF cannot guarantee us a satisfactory detection result.
- The detection performance of AE is just slightly better than LOF, but is worse than the other 4 methods. This validates that one single autoencoder is not enough to capture the correlations of the features of the normal data.
- IF and OCSVM have similar detection performances, whereas LS performs slightly worse than them. However, as there are many hyper-parameters in OCSVM, it is very difficult to select the best hyper-parameters for the implementation. During the training process of IF, there is some randomness in IF. Therefore, for the data of some labels (e.g. 'bird'), IF performs badly.
- ADAE outperforms the other detection methods for all the categories. As can be

seen in Table 5.2, ADAE achieves the highest average AUC score and F1 score.

Here, the precision-recall curves of these detection algorithms evaluated on the STL-10 aircraft dataset are presented in Fig. 5.5. The precision-recall curve of ADAE totally covers the other precision-recall curves. Under the same detection precision, the proposed ADAE algorithm can recall much more image outliers.

Figure 5.5: Precision-Recall Curves evaluated on STL-10 aircraft dataset

Table 5.3 also presents the comparison results on Caltech-101, Scene_data and ImageNet in terms of Precision, Recall and F1 scores. The scores are averaged over all the categories for each dataset. ADAE also outperforms other methods on these datasets.

Figure 5.6: AUC score comparisons on each category in STL-10 dataset

Figure 5.7: F1 score comparisons on each category in STL-10 dataset

5.3.5 Evaluation of ADAE with SGL constraint (ADAE-SGL)

The author also evaluates the detection performance of ADAE-SGL in this subsection. At the beginning, the Pareto-based learning framework presented in Fig. 5.2 is adopted to determine the penalty factors of SGL (λ and μ). Then the detection performance of ADAE-SGL is compared with that of ADAE.

Pareto Front

Pareto-based learning framework is utilized to obtain λ and μ by testing on "airplane" data in STL-10 dataset. Fig. 5.8 presents the Pareto Front which composed of 40 solutions.

For most multi-objective optimization problem, the knee point solution is preferred

Figure 5.8: Pareto Front obtained for the "airplane" image data in STL-10 composed of 40 solutions

if no other user-specific or problem-specific preferences are available. Based on the Pareto Front in Fig. 5.8, the best compromise solution is selected at the knee point: $\lambda = 0.00563$ and $\mu = 0.9289253$. Therefore, in the following evaluations, the author utilizes this set of λ and μ as the penalty factors of SGL for each autoencoder in ADAE-SGL.

Comparison with ADAE

By evaluating on Caltech-101, Scene_data, STL-10, and ImageNet datasets, the author compares the detection results in terms of recall, precision and F1 score. In addition, two sparse evaluation metrics are also introduced to reveal the power of SGL: Neuro Sparsity and Weight Sparsity. The Neuro Sparsity is defined as the number of zeroed-out neurons over the total neuron number, whereas the Weight Sparsity is the

Table 5.4: Detection Results of ADAE

	Precision	Recall	F1	Neuro Sparsity	Weight Sparsity
Caltech-101	1.0000	1.0000	1.0000	100.00%	100.00%
Scene_data	0.9722	0.9780	0.9751	100.00%	100.00%
STL-10	0.9216	0.9441	0.9299	100.00%	100.00%
ImageNet	0.8709	0.9401	0.9050	100.00%	100.00%

Table 5.5: Detection Results of ADAE-SGL

	Precision	Recall	F1	Neuro Sparsity	Weight Sparsity
Caltech-101	1.0000	1.0000	1.0000	40.10%	44.45%
Scene_data	0.9741	0.9760	0.9750	39.37%	45.45%
STL-10	0.9109	0.9411	0.9252	39.79%	44.00%
ImageNet	0.8651	0.9442	0.9029	37.98%	41.49%

proportion of weight parameters with zero values.

Table 5.4 and Table 5.5 present the comparisons of ADAE and ADAE-SGL. As can be seen, ADAE-SGL achieves similar or slightly worse detection performance than ADAE on all the datasets while only using 60% active neurons. It should be noted that the Weight Sparsity is larger than the Neuro Sparsity. This reveals that the SGL constraint not only generates some inactive neurons, but also generates the sparse solutions of weights among the remaining active neurons.

5.4 Conclusions

In this chapter, the author addresses the problem of outlier detection in high-dimensional image datasets. Firstly the author combines autoencoder with Adaboost to capture the statistical correlations among normal data's features. Moreover, the author introduces the Sparse Group Lasso constraint to every autoencoder in order to obtain a compact model. The author also proposes Adagrad Proximal Gradient Descent to optimize the new training objective, and utilizes Pareto-based learning framework to obtain the best penalty factors of SGL. By evaluating on several image

datasets, the proposed detection algorithm not only has a compact detection model, but can also detect the image outliers efficiently.

Chapter 6

Conclusions and Future Work

In this chapter, the author summarizes the thesis by examining the benefits and trade-offs of the proposals and sketching out the directions for future work.

6.1 Conclusion

6.1.1 Detection of Network Anomalies using MSPCA with Sketches

In Chapter 3, the author improves the Conventional-MSPCA anomaly detection algorithm. The Conventional-MSPCA attempts to utilize PCA to filter every scale of the data matrix's wavelet coefficient, so that it can fully separate the normal and abnormal data fractions. However, multiple PCA calculations result in high time complexity in the Conventional-MSPCA algorithm. Therefore, the author needs to resolve this challenge.

The author has found that most energy of the anomalies is hidden in the high

frequency matrices, whereas the low frequency domain mainly contains more information of the normal traffic. Thus, there is no need to perform PCA filter to the high frequency wavelet matrices. The author proposes the MMECE threshold to control the PCA filter level. Furthermore, for the low frequency wavelet matrices, the author adopts the Bayesian-PCA to filter them. Bayesian-PCA has the benefits of low time complexity and automatic selection of PC numbers. Therefore, by introducing the MMECE threshold and the Bayesian-PCA, the proposed improved-MSPCA is able to separate subspaces efficiently with lower time complexity.

The author also combines improved-MSPCA algorithm with sketch data structure. This sketch-MSPCA network anomaly detection system is able to identify the anomalous source IP addresses. The results show the good anomaly detection performance of the proposed detection system.

6.1.2 Power Spectrum Entropy based Detection and Mitigation of Low-Rate DoS Attacks

In Chapter 4, the author studies detection and mitigation of LDoS attacks in the bottleneck routers. As LDoS attacks send periodic pulsing packets to the routers, it becomes challenging to detect the LDoS attacks by using NIDS of the network servers. It is necessary to detect the LDoS attacks in the bottleneck router side.

The author has introduced a novel LDoS detection metric - PSE which combines signal processing technique with the information entropy. Due to the periodic property of the LDoS attack, most of its energy is located at the low frequency domain, which results in small value of PSE. For the normal TCP flow, the corresponding PSE value is larger than that of LDoS attacks. Therefore, for any incoming network flow, if its PSE falls below a threshold, it will be treated as an LDoS attack.

The author has also proposed a LDoS mitigation algorithm: PRRED queuing algorithm which combines PSE with Robust RED queuing algorithm. There are two detection phases in PRRED. The first one is a rough detection phase which utilizes the packet arrival time to roughly detect the suspicious packets. Then, these suspicious packets will be passed to the next detection phase to calculate their PSE values. Based on the PSE values, the algorithm will determine whether to drop the packet or not.

6.1.3 Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection

In Chapter 5, the author addresses the image outlier detection problem. Traditional autoencoder based outlier detection method cannot learn the total correlations of the normal images, as they just adopt one simple autoencoder. The author combines autoencoder with Adaboost algorithm. With the assistance of Adaboost, ADAE algorithm is able to fully capture the feature correlations of the normal images so as to increase the detection accuracy.

Furthermore, the author has also introduced SGL regularization term for the purpose of obtaining a compact detection model. In order to optimize the SGL term, the author combines Adagrad with Proximal Gradient Descent. The author also purposes the EMO based optimization framework to get the optimal penalty factors of SGL.

6.2 Future Research Directions

For the network anomaly detection, the proposed sketch-MSPCA anomaly detection system is an off-line detection system. Therefore, the future work can focus on developing an online network anomaly detection system. Online PCA algorithm [66] can be

taken here as a reference to modify MSPCA algorithm to the online form.

For the detection and mitigation of LDoS attacks, the attackers may raise some other kinds of LDoS-related attacks. Some attacks may not use a fixed attack period T . Some attackers could randomly draw within some relevant interval to vary the burst duration and attack rate. These could result in the same Denial of Service effect, but would be much harder to detect using power spectrum information. Therefore, a possible direction for future work is to examine the performance of the proposed approach for an attacker employing such strategies. In addition, the author just evaluates the proposed algorithm on NS-3 simulations and public network traces. Testing the PRRED algorithm on a real testbed could be a future work. The Field-Programmable Gate Array (FPGA) can be utilized here to implement Fourier and Wavelet Analysis in the hardware. Therefore, with the help of FPGA, PRRED algorithm can be embedded into the CISCO routers to mitigate LDoS attacks in real time.

In the image outlier detection problem, for every autoencoder in ADAE algorithm, the author just adopts the reconstruction loss. The normal autoencoder and the reconstruction loss are sensitive to the noise. In the future work, the Laplacian pyramid loss [73] can be used here to improve the proposed outlier detection algorithm. Furthermore, denoising autoencoder [110] can be applied to the algorithm to make the detection model robust to the noise.

Currently, the author has investigated the anomaly/outlier detection in 1D dataset (network anomaly detection) and 2D dataset (image outlier detection). Anomaly detection also can be applied to the 3D dataset, for example anomaly detection in the videos of crowded scenes. Therefore, crowded scenes anomaly detection could be another future direction. As crowded scenes may contain temporal and spatial abnormalities, temporal anomaly detection and spatial anomaly detection both need to be considered. The sequence-to-sequence (seq2seq) model may be a potential method to

address this problem. Seq2seq models [105] [27] have achieved great success in a variety of natural language processing tasks, such as machine translation and text summarization. Similar to PCA and autoencoder, the reconstruction error loss function can be applied to learn the seq2seq model, so that the model can capture the spatial and temporal correlations among the crowded scenes. The reconstruction errors can later be adopted to determine the abnormality of each frame in crowded scenes.

Author's Publications

Journals

- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee, Chiew Tong Lau and Yaochu Jin, “Evolutionary Multi-objective Optimization of Ensemble Autoencoder for Image Outlier Detection ”, *Neurocomputing* 309 (2018): 192-200.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “Power Spectrum Entropy based Detection and Mitigation of Low-Rate DoS Attacks ”, *Computer Networks* 136 (2018): 80-94.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “Detection of Network Anomalies using Improved- MSPCA with Sketches ”, *Computers & Security* 65 (2017): 314-328.

Conferences

- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “Autoencoder based Network Anomaly Detection ”, in *Proceedings of Wireless Telecommunication Symposium (WTS)*, Phoenix, USA, pp. 1–5, Apr 2018.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “FRRED: Fourier robust RED algorithm to detect and mitigate LDoS attacks ”, in *publication description Zooming Innovation in Consumer Electronics International Conference (ZINC)*, Novi Sad, Serbia, pp. 13-17, May 2017.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “Combining MIC feature selection and feature-based MSPCA for network traffic anomaly detection”, in *Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC), 2016 Third International Conference on, IEEE.*, Moscow, Russia, pp. 176-181, July 2016.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “A MSPCA based intrusion detection algorithm for detection of DDoS attack ”, in *Communications in China (ICCC), 2015 IEEE/CIC International Conference on*, Shenzhen, China, pp. 1-5, Nov 2015.
- **Zhaomin Chen**, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, “A novel anomaly detection system using feature-based MSPCA with sketch ”, in *Wireless and Optical Communication Conference (WOCC)*, Newark, USA, pp. 1-6, April 2015.

Bibliography

- [1] Tarem Ahmed, Mark Coates, and Anukool Lakhina. Multivariate online anomaly detection using kernel recursive least squares. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 625–633. IEEE, 2007.
- [2] Arindom Ain, Monowar H Bhuyan, Dhruva K Bhattacharyya, and Jugal K Kalita. Rank correlation for low-rate ddos attack detection: An empirical evaluation. *IJ Network Security*, 18(3):474–480, 2016.
- [3] Mark Allman and Vern Paxson. On estimating end-to-end network path properties. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 263–274. ACM, 1999.
- [4] Jerone TA Andrews, Edward J Morton, and Lewis D Griffin. Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1):21, 2016.
- [5] Muhammad H Arshad and Philip K Chan. Identifying outliers via clustering for anomaly detection. 2003.
- [6] Bhavik R Bakshi. Multiscale pca with application to multivariate statistical process monitoring. *AIChE journal*, 44(7):1596–1610, 1998.
- [7] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 71–82. ACM, 2002.
- [8] M.H. Bhuyan, A. Kalwar, A. Goswami, D.K. Bhattacharyya, and J.K. Kalita. Low-rate and high-rate distributed DoS attack detection using partial rank correlation. In *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, apr 2015.

-
- [9] Monowar H Bhuyan, DK Bhattacharyya, and Jugal K Kalita. An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection. *Pattern Recognition Letters*, 51:1–7, 2015.
 - [10] Christopher M Bishop. Bayesian pca. *Advances in neural information processing systems*, pages 382–388, 1999.
 - [11] Pierre Borgnat, Guillaume Dewaele, Kensuke Fukuda, Patrice Abry, and Kenjiro Cho. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM 2009, IEEE*, pages 711–719. IEEE, 2009.
 - [12] Daniela Brauckhoff, Kave Salamatian, and Martin May. Applying pca for traffic anomaly detection: Problems and solutions. In *INFOCOM 2009, IEEE*, pages 2866–2870. IEEE, 2009.
 - [13] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.
 - [14] Jake D Brutlag. Aberrant behavior detection in time series for network monitoring. In *LISA*, volume 14, pages 139–146, 2000.
 - [15] Joel Brynielsson and Rishie Sharma. Detectability of low-rate http server dos attacks using spectral analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 954–961. IEEE, 2015.
 - [16] Joel Brynielsson and Rishie Sharma. Detectability of low-rate http server dos attacks using spectral analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 954–961. IEEE, 2015.
 - [17] Christian Callegari, Loris Gazzarrini, Stefano Giordano, Michele Pagano, and Teresa Pepe. A novel pca-based network anomaly detection. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.
 - [18] Christian Callegari, Stefano Giordano, Michele Pagano, and Teresa Pepe. Wave-cusum: Improving cusum performance in network anomaly detection by means of wavelet analysis. *Computers & Security*, 31(5):727–735, 2012.
 - [19] José Camacho, Alejandro Pérez-Villegas, Pedro García-Teodoro, and Gabriel Maciá-Fernández. Pca-based multivariate statistical network monitoring for anomaly detection. *Computers & Security*, 59:118–137, 2016.

- [20] Diego Carrera, Giacomo Boracchi, Alessandro Foi, and Brendt Wohlberg. Detecting anomalous structures by convolutional sparse models. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015.
- [21] Philip K Chan, Wei Fan, Andreas L Prodromidis, and Salvatore J Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, 14(6):67–74, 1999.
- [22] Xi-ping CHEN, Hai-jie MAO, and Wei LI. Study on choosing mother wavelet for signal singularity detection based on matlab [j]. *Computer Simulation*, 11:015, 2004.
- [23] Yu Chen and Kai Hwang. Collaborative detection and filtering of shrew ddos attacks using spectral analysis. *Journal of Parallel and Distributed Computing*, 66(9):1137–1151, 2006.
- [24] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee Francis, and Chiew Tong Lau. A mspca based intrusion detection algorithm tor detection of ddos attack. In *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–5. IEEE, 2015.
- [25] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee Francis, and Chiew Tong Lau. Combining mic feature selection and feature-based mspca for network traffic anomaly detection. In *2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, pages 176–181. IEEE, 2016.
- [26] Roman Chertov, Sonia Fahmy, and Ness B Shroff. Emulation versus simulation: A case study of tcp-targeted denial of service attacks. In *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, pages 10–pp. IEEE, 2006.
- [27] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [28] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 48. ACM, 2009.

- [29] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.
- [30] Christophe Croux, Peter Filzmoser, and Maria Rosario Oliveira. Algorithms for projection–pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225, 2007.
- [31] Alberto Dainotti, Antonio Pescapé, and Giorgio Ventre. Nis04-1: Wavelet-based detection of dos attacks. In *IEEE Globecom 2006*, pages 1–6. IEEE, 2006.
- [32] Marco De Vivo, Eddy Carrasco, Germinal Isern, and Gabriela O de Vivo. A review of port scanning techniques. *ACM SIGCOMM Computer Communication Review*, 29(2):41–48, 1999.
- [33] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [35] Guillaume Dewaele, Kensuke Fukuda, Pierre Borgnat, Patrice Abry, and Kenjiro Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. In *Proceedings of the 2007 workshop on Large scale attack defense*, pages 145–152. ACM, 2007.
- [36] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- [37] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934, 2009.
- [38] Ambedkar Dukkipati, Debarghya Ghoshdastidar, and Jinu Krishnan. Mixture modeling with compact support distributions for unsupervised learning. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2706–2713. IEEE, 2016.
- [39] Meriem El Azami, Carole Lartizien, and Stéphane Canu. Converting svdd scores into probability estimates: Application to outlier detection. *Neurocomputing*, 268:64–75, 2017.

- [40] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.
- [41] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer, 2000.
- [42] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.
- [43] Anja Feldmann, Anna C Gilbert, and Walter Willinger. Data networks as cascades: Investigating the multifractal nature of internet wan traffic. In *ACM SIGCOMM Computer Communication Review*, volume 28, pages 42–55. ACM, 1998.
- [44] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, 1(4):397–413, 1993.
- [45] Romain Fontugne, Pierre Borgnat, Patrice Abry, and Kensuke Fukuda. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *Proceedings of the 6th International Conference*, page 8. ACM, 2010.
- [46] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.
- [47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [48] Yu Gu, Andrew McCallum, and Don Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 32–32. USENIX Association, 2005.
- [49] Mina Guirguis, Azer Bestavros, and Ibrahim Matta. Exploiting the transients of adaptation for roq attacks on internet resources. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 184–195. IEEE, 2004.

- [50] Sara Hakami, Zainab Zaidi, Bjorn Landfeldt, and Tim Moors. Detection and identification of anomalies in wireless mesh networks using principal component analysis (pca). In *2008 International Symposium on Parallel Architectures, Algorithms, and Networks (i-span 2008)*, pages 266–271. IEEE, 2008.
- [51] Thomas R Henderson, Mathieu Lacage, George F Riley, C Dowell, and J Kopena. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 15:17, 2008.
- [52] Yosuke Himura, Kensuke Fukuda, Kenjiro Cho, and Hiroshi Esaki. An automatic and dynamic parameter tuning of a statistics-based anomaly detection algorithm. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.
- [53] Wenjie Hu, Yihua Liao, and V Rao Vemuri. Robust anomaly detection using support vector machines. In *Proceedings of the international conference on machine learning*, pages 282–289, 2003.
- [54] Chin-Tser Huang, Sachin Thareja, and Yong-June Shin. Wavelet-based real time detection of network traffic anomalies. In *Securecomm and Workshops, 2006*, pages 1–7. IEEE, 2006.
- [55] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.
- [56] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.
- [57] Ruoyi Jiang, Hongliang Fei, and Jun Huan. A family of joint sparse pca algorithms for anomaly localization in network data streams. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2421–2433, 2013.
- [58] Taisong Jin, Zhiling Liu, Zhengtao Yu, Xiaoping Min, and Lingling Li. Locality preserving collaborative representation for face recognition. *Neural Processing Letters*, 45(3):967–979, 2017.
- [59] Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008.

- [60] Yoshiki Kanda, Romain Fontugne, Kensuke Fukuda, and Toshiharu Sugawara. Admire: Anomaly detection method using entropy-based pca with three-step sketches. *Computer Communications*, 36(5):575–588, 2013.
- [61] Yoshiki Kanda, Kensuke Fukuda, and Toshiharu Sugawara. Evaluation of anomaly detection based on sketch and pca. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.
- [62] Nikolaos Kouiroukidis and Georgios Evangelidis. The effects of dimensionality curse in high dimensional knn search. In *Informatics (PCI), 2011 15th Panhellenic Conference on*, pages 41–45. IEEE, 2011.
- [63] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. LoOP: local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1649–1652, 2009.
- [64] Aleksandar Kuzmanovic and Edward W Knightly. Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 75–86. ACM, 2003.
- [65] Aleksandar Kuzmanovic and Edward W Knightly. Low-rate tcp-targeted denial of service attacks and counter strategies. *IEEE/ACM Transactions on Networking (TON)*, 14(4):683–696, 2006.
- [66] Dima Kuzmin and Manfred K Warmuth. Online kernel pca with entropic matrix updates. In *Proceedings of the 24th international conference on Machine learning*, pages 465–472. ACM, 2007.
- [67] Yu-Kwong Kwok, Rohit Tripathi, Yu Chen, and Kai Hwang. Hawk: Halting anomalies with weighted choking to rescue well-behaved tcp sessions from shrew ddos attacks. In *Networking and mobile computing*, pages 423–432. Springer, 2005.
- [68] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 219–230. ACM, 2004.
- [69] Aleksandar Lazarevic, Vipin Kumar, and Jaideep Srivastava. Intrusion detection: A survey. In *Managing Cyber Threats*, pages 19–78. Springer, 2005.
- [70] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer*

- vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.
- [71] Xin Li, Fang Bian, Mark Crovella, Christophe Diot, Ramesh Govindan, Gianluca Iannaccone, and Anukool Lakhina. Detection and identification of network anomalies using sketch subspaces. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 147–152. ACM, 2006.
- [72] Yang Li, Binxing Fang, Li Guo, and You Chen. Network anomaly detection based on tcm-knn algorithm. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 13–19. ACM, 2007.
- [73] Haibin Ling and Kazunori Okada. Diffusion distance for histogram comparison. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 246–253. IEEE, 2006.
- [74] Richard Lippmann, Joshua W Haines, David J Fried, Jonathan Korba, and Kumar Das. The 1999 darpa off-line intrusion detection evaluation. *Computer networks*, 34(4):579–595, 2000.
- [75] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008.
- [76] Haiqin Liu and Min Sik Kim. Real-time detection of stealthy ddos attacks using time-series decomposition. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.
- [77] Wei Liu, Gang Hua, and John R Smith. Unsupervised one-class learning for automatic outlier removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014.
- [78] Yang Liu, Linfeng Zhang, and Yong Guan. Sketch-based streaming pca algorithm for network-wide traffic anomaly detection. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 807–816. IEEE, 2010.
- [79] Mohammad Ahamdi Livani and Mahdi Abadi. Distributed pca-based anomaly detection in wireless sensor networks. In *Internet Technology and Secured Transactions (ICITST), 2010 International Conference for*, pages 1–8. IEEE, 2010.
- [80] Wei Lu and Ali A Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing*, 2009:4, 2009.

- [81] Jingtang Luo, Xiaolong Yang, Jin Wang, Jie Xu, Jian Sun, and Keping Long. On a mathematical model for low-rate shrew ddos. *IEEE Transactions on Information Forensics and Security*, 9(7):1069–1083, 2014.
- [82] Xiapu Luo and Rocky KC Chang. On a new class of pulsing denial-of-service attacks and the defense. In *NDSS*, 2005.
- [83] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- [84] Snehal A Mulay, PR Devale, and GV Garje. Intrusion detection system using support vector machine and decision tree. *International Journal of Computer Applications*, 3(3):40–43, 2010.
- [85] Amuthan Prabakar Muniyandi, R Rajeswari, and R Rajaram. Network anomaly detection by cascading k-means clustering and c4. 5 decision tree algorithm. *Procedia Engineering*, 30:174–182, 2012.
- [86] George Nychis, Vyas Sekar, David G Andersen, Hyong Kim, and Hui Zhang. An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 151–156. ACM, 2008.
- [87] Francesco Palmieri and Ugo Fiore. Network anomaly detection through nonlinear analysis. *Computers & Security*, 29(7):737–755, 2010.
- [88] Cláudia Pascoal, M Rosario de Oliveira, Rui Valadas, Peter Filzmoser, Paulo Salvador, and António Pacheco. Robust feature selection and robust pca for internet traffic anomaly detection. In *INFOCOM, 2012 Proceedings IEEE*, pages 1755–1763. IEEE, 2012.
- [89] Vern Paxson, Mark Allman, Jerry Chu, and Matt Sargent. Computing tcp’s retransmission timer. Technical report, 2011.
- [90] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.
- [91] John A Quinn and Masashi Sugiyama. A least-squares approach to anomaly detection in static and sequential data. *Pattern Recognition Letters*, 40:36–40, 2014.

- [92] Manikantan Ramadas, Shawn Ostermann, and Brett Tjaden. Detecting anomalous network traffic with self-organizing maps. In *International Workshop on Recent Advances in Intrusion Detection*, pages 36–54. Springer, 2003.
- [93] Haakon Ringberg, Augustin Soule, Jennifer Rexford, and Christophe Diot. Sensitivity of pca for traffic anomaly detection. In *ACM SIGMETRICS Performance Evaluation Review*, volume 35, pages 109–120. ACM, 2007.
- [94] Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shinghon Lau, Satish Rao, Nina Taft, and JD Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 1–14. ACM, 2009.
- [95] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM, 2014.
- [96] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.
- [97] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.
- [98] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- [99] Philipp Seeböck, Sebastian Waldstein, Sophie Klimscha, Bianca S Gerendas, René Donner, Thomas Schlegl, Ursula Schmidt-Erfurth, and Georg Langs. Identifying and categorizing anomalies in retinal imaging data. *arXiv preprint arXiv:1612.00686*, 2016.
- [100] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIG-MOBILE mobile computing and communications review*, 5(1):3–55, 2001.
- [101] Amey Shevtekar, Karunakar Anantharam, and Nirwan Ansari. Low rate tcp denial-of-service attack detection at edge routers. *IEEE Communications Letters*, 9(4):363–365, 2005.

- [102] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [103] Augustin Soule, Kavé Salamatian, and Nina Taft. Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 31–31. USENIX Association, 2005.
- [104] Haibin Sun, John CS Lui, and David KY Yau. Defending against low-rate tcp attacks: dynamic detection and protection. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pages 196–205. IEEE, 2004.
- [105] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [106] Min Tan, Gang Pan, Yueming Wang, Yuting Zhang, and Zhaohui Wu. L1-norm latent svm for compact features in object detection. *Neurocomputing*, 139:56–64, 2014.
- [107] Dan Tang, Kai Chen, XiaoSu Chen, HuiYu Liu, and Xinhua Li. Adaptive ewma method based on abnormal network traffic for ldos attacks. *Mathematical Problems in Engineering*, 2014, 2014.
- [108] Yajuan Tang, Xiapu Luo, Qing Hui, and Rocky KC Chang. Modeling the vulnerability of feedback-control based internet services to low-rate dos attacks. *IEEE Transactions on Information Forensics and Security*, 9(3):339–353, 2014.
- [109] Jan Verbesselt, Achim Zeileis, and Martin Herold. Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*, 123:98–108, 2012.
- [110] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.
- [111] Mark N Wegman and J Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of computer and system sciences*, 22(3):265–279, 1981.
- [112] Zhi-jun Wu, Jin Lei, Di Yao, Ming-Hua Wang, and Sarhan M Musa. Chaos-based detection of ldos attacks. *Journal of Systems and Software*, 86(1):211–221, 2013.

- [113] Zhi-Jun Wu and Meng Yue. Detection of lddos attack based on kalman filtering. *Acta Electronica Sinica*, 36(8):1590e4, 2008.
- [114] Zhijun Wu, Minxiao Wang, Changcan Yan, and Meng Yue. Low-rate dos attack flows filtering based on frequency spectral analysis. *China Communications*, 14(6):98–112, 2017.
- [115] Zhijun Wu, Liyuan Zhang, and Meng Yue. Low-rate DoS attacks detection based on network multifractal. *IEEE Transactions on Dependable and Secure Computing*, 13(5):559–567, sep 2016.
- [116] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015.
- [117] Yang Xiang, Ke Li, and Wanlei Zhou. Low-rate ddos attacks detection and traceback by using new information metrics. *IEEE Transactions on Information Forensics and Security*, 6(2):426–437, 2011.
- [118] Shengkun Xie, Feng Jin, Sridhar Krishnan, and Farook Sattar. Signal feature extraction by multi-scale pca and its application to respiratory sound classification. *Medical & biological engineering & computing*, 50(7):759–768, 2012.
- [119] Shengkun Xie, Pietro Lio, and Anna T Lawniczak. A comparative study of noise effect on wavelet based de-noising methods. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pages 919–926. IEEE, 2009.
- [120] Chen Xiping, Mao Haijie, and Li Wei. Study on choosing mother wavelet for signal singularity detection based on matlab. *Computer Simulation*, 21(11):48–67, 2004.
- [121] Guo Yi, Yan Juwei, Qiu Han, and Zhang Liancheng. A crf-theory-based method for bgp-ldos attack detection. In *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on*, pages 1071–1075. IEEE, 2016.
- [122] Joseph P Zbilut, Alessandro Giuliani, and Charles L Webber Jr. Recurrence quantification analysis and principal components in the detection of short complex signals. *arXiv preprint chao-dyn/9712017*, 1997.
- [123] Bin Zhang, Jiahai Yang, Jianping Wu, Donghong Qin, and Lei Gao. Pca-subspace method—Is it good enough for network-wide anomaly detection. In *2012 IEEE Network Operations and Management Symposium*, pages 359–367. IEEE, 2012.

-
- [124] Changwang Zhang, Zhiping Cai, Weifeng Chen, Xiapu Luo, and Jianping Yin. Flow level detection and filtering of low-rate ddos. *Computer Networks*, 56(15):3417–3431, 2012.
 - [125] Changwang Zhang, Jianping Yin, Zhiping Cai, and Weifeng Chen. Rred: robust red algorithm to counter low-rate denial-of-service attacks. *IEEE Communications Letters*, 14(5):489–491, 2010.
 - [126] Min-Ling Zhang. ML-RBF:RBF neural networks for multi-label learning. *Neural Processing Letters*, 29(2):61–74, 2009.
 - [127] Hao Zheng, Jianchun Xie, and Zhong Jin. Heteroscedastic sparse representation based classification for face recognition. *Neural processing Letters*, 35(3):233–244, 2012.
 - [128] Wu Zhi-Jun, Zhang Hai-Tao, Wang Ming-Hua, and Pei Bao-Song. Msabms-based approach of detecting ldos attack. *computers & security*, 31(4):402–417, 2012.