REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
24-07-2017	Final Report	25-Apr-2013 - 24-Apr-2017

4. TITLE AND SUBTITLE

Final Report: Designing a Robust Closed-Loop Intrusion Detection Predictive Model Using Signal Processing Techniques in Cloud Computing Environment

5a. CONTRACT NUMBER	W911NF-13-1-0143
5b. GRANT NUMBER	
5c. PROGRAM ELEMENT NUMBER	206022
5d. PROJECT NUMBER	
5e. TASK NUMBER	
5f. WORK UNIT NUMBER	

6. AUTHORS

Soo-Yeon Ji, seonho Choi, DongHyun Jeong

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

Bowie State University
14000 Jericho Park Road

Bowie, MD 20715 -9465

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
62702-CS-REP.26

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

As network attacks become more prevalent and complex, it is inevitable to find efficient ways to protect our computing infrastructures. Recently, researchers have begun to harness both machine learning and cloud computing technology to identify threats with reducing the overall computation time of detecting them. The objective of this research is to design an intrusion detection (ID) predictive model to identify abnormal network behaviors (i.e. abnormalities). Advanced signal processing techniques are utilized to design the model. With the model, it would be feasible to protect corporate and government agency's computing infrastructures and data securely. Specifically,

15. SUBJECT TERMS

Network intrusion detection;signal processing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Soo-Yeon Ji
UU	UU	UU	UU		19b. TELEPHONE NUMBER 301-860-4458

Report Title

Final Report: Designing a Robust Closed-Loop Intrusion Detection Predictive Model Using Signal Processing Techniques in Cloud Computing Environment

ABSTRACT

As network attacks become more prevalent and complex, it is inevitable to find efficient ways to protect our computing infrastructures. Recently, researchers have begun to harness both machine learning and cloud computing technology to identify threats with reducing the overall computation time of detecting them. The objective of this research is to design an intrusion detection (ID) predictive model to identify abnormal network behaviors (i.e. abnormalities). Advanced signal processing techniques are utilized to design the model. With the model, it would be feasible to protect corporate and government agency's computing infrastructures and data securely. Specifically, this research focuses on 1) extracting significant features that represent the characteristics of abnormal behaviors by applying the signal processing techniques, 2) generating a predictive model to determine and differentiate various attacks (DoS, Probe, and R2L), 3) utilization of a visual analytic tool to identify relationship among the features, and 4) exploring current research trends and directions in network intrusion detection by examining innovative network intrusion detection approaches that utilize both machine learning algorithms and cloud computing technologies. This research is conducted mainly at Bowie State University. The University of the District of Columbia (UDC) joins this project as a sub-awardee.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
07/21/2017	2	Soo-Yeon Ji, Seonho Choi, Dong. H. Jeong. Designing an Internet Traffic Predictive Model by applying a Signal Processing, Journal of Network and System Management, (04 2014): 998. doi: 332,784.00
07/21/2017	21	Benjamin Simeon Harvey, Soo-Yeon Ji. Cloud-Scale Genomic Signals Processing for Robust Large-Scale Cancer Genomic Microarray Data Analysis, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, (01 2017): 238. doi: 1,050,975.00
07/21/2017	20	Dong Hyun Jeong,Soo-Yeon Ji, Evan A Suma, Byunggu Yu, Remco Chang. Designing a collaborative visual analytics system to support users' continuous analytical processes, Human-centric Computing and Information Sciences, (02 2015): 5. doi: 1,050,974.00
07/21/2017	14	Soo-Yeon Ji,Bong-Keun Jeong, Seonho Choi, Dong. H. Jeong. A Multi-level Intrusion Detection Method for AbnormalNetwork Behaviors, Journal of Network and Computer Applications (Elsevier), (08 2015): 9. doi: 364,918.00
07/21/2017	12	Nathan Keegan, Soo-Yeon Ji, Aastha Chaudhary, Claude Concolato, Byungu Yu, DongHyun Jeong. A Survey of Cloud-based Network Intrusion Detection Analysis, Human-centric Computing and information Science, (04 2015): 19. doi: 364,913.00
08/26/2015	8	Soo-Yeon Ji , Seonho Choi , Dong Hyun Jeong. Designing an Internet Traffic PredictiveModel by Applying a Signal ProcessingMethod, Journal of Network and System Management, (09 2014): 1. doi: 364,901.00
08/26/2015	9	Dong Hyun Jeong, Soo-Yeon Ji, Evan A Suma, Byunggu Yu, Remco Chang. Designing a collaborative visual analyticssystem to support users' continuous analyticalprocesses, Other, (02 2015): 1. doi: 364,905.00
08/26/2016	15	Soo-Yeon Ji, Bong-Keun Jeong, Seonho Choi, Dong. H. Jeong. A multi-level intrusion detection method for abnormal network behaviors, Journal of Network and Computer Applications, (02 2016): 9. doi: 1,014,705.00
TOTAL:	**8**	

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Three posters has been presented from Apr. 2013 to Apr. 2017

Number of Presentations: 3.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
07/21/2017	22	DongHyun Jeong, Bong-Kyun Jeong, Soo-Yeon Ji. Designing a Hybrid Approach with Computational Analysis and Visual Analytics to Detect Network Intrusions, Computing and Communication Workshop and Conference (CCWC). 09-JAN-17, Las Vegas, NV, USA. : ,
08/28/2016	17	Soo-Yeon Ji, Seonho Choi, Dong Hyun Jeong. Designing a Two-Level Monitoring Method to Detect Network Abnormal Behaviors, IEEE IRI 2014. 14-AUG-14, San Francisco, California, USA. : ,
08/28/2016	16	LASSINE CHERIF, SOO-YEON JI, DONG HYUN JEONG. Utilization of visual analytical approach to detect anomalies in large network traffic data, NIST Data Science Symposium Proceedings. 04-MAR-14, Gaithersburg, MD. : ,
08/28/2016	18	Dong Hyun Jeong, Soo-Yeon Ji, Tera Greensmith, Byunggu Yu, Remco Chang. Understanding Implicit and Explicit Interface Tools to Perform Visual Analytics Tasks, IEEE International Conference on Information Reuse and Integration (IRI). 14-AUG-14, San Francisco, California, USA. : ,
08/28/2016	19	Benjamin Harvey, Soo-Yeon Ji. Cloud-Scale Genomic Signals Processing Classification Analysis for Gene Expression Microarray Data, IEEE International Conference of the Engineering in Medicine and Biology Society. 27-AUG-14, Chicago, IL. : ,
TOTAL:	**5**	

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
08/26/2016	10.00	Soo-Yeon Ji, Seonho Choi, Dong H. Jeong. Designing a Two-Level Monitoring Method toDetect Network Abnormal Behaviors, IEEE International Conference on Information Reuse and Integration (IRI). 13-AUG-14, San Francisco, California, USA. : ,
08/26/2016	11.00	Dong Hyun Jeong, Soo-Yeon Ji,Tera Greensmith,Byunggu Yu,Remco Chang. Understanding Implicit and Explicit Interface Toolsto Perform Visual Analytics Tasks, IEEE International Conference on Information Reuse and Integration (IRI). 13-AUG-14, San Francisco, CA, USA.. : ,
08/26/2016	13.00	Benjamin Harvey, Soo-Yeon Ji. Cloud-Scale Genomic Signals Processing Classification Analysis forGene Expression Microarray Data, IEEE Engineering in Medicine and Biology Society (EMBC). 26-AUG-14, Chicago, Illinois, USA.. : ,
TOTAL:		**3**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper
TOTAL:	

Number of Manuscripts:

Books

Received	Book
TOTAL:	

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

NA

Graduate Students

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Post Doctorates

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	1.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:	1.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	1.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See the attached file

Technology Transfer

NA

2017 FINAL REPORT

GRANT NO: W911NF-13-1-0143 (62702-CS-REP)

PROGRAM MANAGER: DR. CLIFF X. WANG

Proposal Title:

Designing a Robust Closed-loop Intrusion Detection Predictive Model Using Signal Processing Techniques in Cloud Computing Environment

Principal Investigator: Soo-Yeon Ji
Bowie State University
Computer Science Department
14000 Jericho Park Rd.
Bowie, MD 20715
sji@bowiestate.edu
Office) 301-860-4458

Final Report Submission date: July 24, 2017

Contents

List of Figures

List of Tables

PROJECT INFORMATION

a) Contract number: W911NF-13-1-0143

b) Period of performance being reported: Aug. 1, 2016 - Apr. 24, 2017

c) Principal Investigator: Soo-Yeon Ji

d) Contracting Officer Representative:

Dr. Cliff X. Wang
DEPARTMENT OF THE ARMY
US ARMY RESEARCH, DEVELOPMENT AND ENGINEERING COMMAND
ARMY RESEARCH OFFICE
Email: cliff.x.wang.civ@mail.mil
Tel: (919) 549-4207

1 SUMMARY

As network attacks become more prevalent and complex, it is inevitable to find efficient ways to protect our computing infrastructures and to understand network traffic patterns. Recently, researchers have begun to harness both machine learning and cloud computing technology to identify threats with reducing the overall computation time of detecting them. The objective of this research is to design an intrusion detection (ID) predictive model to identify abnormal network behaviors (i.e. abnormalities). Advanced signal processing techniques are utilized to design the model. Also, we propose a hybrid approach of integrating computational analysis with visual analytics to detect network intrusions. For the computational analysis, both Multi-Resolution Analysis (MRA) and Principal Component Analysis (PCA) are applied to analyze network traffic data. First, Discrete Wavelet Transform (DWT) is utilized as the MRA approach to extract features from the network traffic data. After determining statistically significant features from a statistical validation, PCA is applied to transform the extracted features for identifying principal components in eigenspace. Lastly, a visual analytics tool is designed to help the user conduct an interactive visual analysis on detected network intrusions by initiating interactive visual validation and verification. Overall, we found that our approach is good for detecting network intrusions as well as to understand their patterns.

Specifically, this research focused on 1) extracting significant features that represent the characteristics of abnormal behaviors by applying the signal processing techniques, 2) generating a predictive model to determine and differentiate various attacks (DoS, Probe, and R2L), 3) utilization of a visual analytic tool to identify relationship among the features, and 4) exploring current research trends and directions in network intrusion detection by examining innovative network intrusion detection approaches that utilize both machine learning algorithms and cloud computing technologies.

This research was conducted mainly at Bowie State University. The University of the District of Columbia (UDC) joined this project as a sub-awardee.

2 PUBLICATIONS

The collaborators (PI, Co-PI, and sub-awardee) have managed regularly scheduled meetings and conference calls to discuss research progress and to identify possible future research directions. Based on research accomplishments, the following papers have been published during the reporting period (Apr. 2013 ~ Apr. 2017).

2.1 Published Journal Papers:

- N.Keegan, **S. Y. Ji**, A. Chaudhary, C. Concolato, B. Yu, D.H. Jeong, A Survey of Cloud-based Network Intrusion Detection Analysis, *Human-centric Computing and Information Sciences* (DOI: 10.1186/s13673-016-0076-z), Dec. 5, 2016
- **S. Y. Ji**, Bong-Keun Jeong, Seonho Choi, Dong. H. Jeong, A multi-level intrusion detection method for abnormal network behaviors, *Journal of Network and Computer Applications*, Vol. 62, Issue C. pp. 9-17, (DOI: 10.1016/j.jnca.2015.12.004) Feb. 2016.
- Benjamin Harvey, **S.Y. Ji**, Cloud-Scale Genomic Signals Processing for Robust Large-Scale Cancer Genomic Microarray Data Analysis, *IEEE Journal of Biomedical and Health Informatics*, No. 99, pp. 238 - 245, (DOI:10.1109/JBHI.2015.2496323), Jan. 2017
- Dong H. Jeong, **S. Y. Ji**, Evan A Suma, Byunggu Yu and Remco Chang, Designing a collaborative visual analytics system to support users' continuous analytical processes, *Human-centric Computing and Information Sciences, Springer*, volume 5, 2015. (DOI:10.1186/s13673-015-0023-4)
- **S. Y. Ji**, Seonho Choi, Dong. H. Jeong, Designing an Internet Traffic Predictive Model by applying a Signal Processing, *Journal of Network and System Management, Springer-Verlag*, Vol. 23, No. 4, (DOI: 10.1007/s10922-014-9335-3), 2014.

2.2 Published Conference / Workshop Papers

- Dong H. Jeong, Bong-Keun Jeong, **S. Y. Ji**, D. H. Jeong, B. K. Jeong and S. Y. Ji, Designing a hybrid approach with computational analysis and visual analytics to detect network intrusions, *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2017, pp. 1-7. doi: 10.1109/CCWC.2017.7868417
- **S. Y. Ji**, Seonho Choi, Dong H. Jeong, Designing a two-level monitoring method to detect network abnormal behaviors, *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, Redwood City, CA, 2014, pp. 703-709. doi: 10.1109/IRI.2014.7051958
- Dong H. Jeong, **S. Y. Ji**, T. Greensmith, B. Yu, R. Chang, Understanding implicit and explicit interface tools to perform visual analytics tasks, *Proceedings of the 2014 IEEE 15th International*

Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, 2014, pp. 687-694. doi: 10.1109/IRI.2014.7051956

2.3 Poster Presentation:

- Jesuye David,**S. Y. Ji**, Designing a computational method for identifying abnormal behaviors on smartphones, The 21st Annual Posters on the Hill, Washington DC, April 25-26, 2017.
- D.H. Jeong, **S. Y. Ji**, Applying Data Transformation to Derive Insights for Network Intrusion Detection, IEEE Symposium on Visualization for Cybersecurity (VizSec 2016), Baltimore, MD, Oct.24, 2016.
- L. Cherif, **S. Y. Ji**, D.H. Jeong, Utilization of visual analytical approach to detect anomalies in large network traffic data, NIST Data Science Symposium, March 4-5, Gaithersburg, MD, 2014.

3 OBJECTIVES

Abnormal network traffic analysis has become an increasingly important research topic of protecting computing infrastructures from intruders. Due to a high-volume of network traffic stream, it is difficult to detect network intrusions precisely. To have a better knowledge about the network intrusions, this project focuses on designing a multi-level network intrusion detection technique by addressing limitations in existing techniques. Since analyzing the network traffic stream often requires high-end computing resources, integration of cloud computing technology is necessary. To apply the technology to our network intrusion detection study, we begin by reviewing known cloud-based network intrusion detection techniques. Mainly, our study consists of three steps as 1) understanding hidden underlying patterns in network traffic data by creating reliable rules to identify network abnormality, 2) generating a predictive model to determine exact attack categories, and 3) integrating a visual analytics tool to conduct an interactive visual analysis and validate the identified attacks, 4) exploring cloud-based network detection techniques. This study mainly focused on the followings;

a) Validating designed feature extraction approaches

b) Designing a multi-level network abnormal model detection method

- Generate reliable rules to detect abnormal network behaviors
- Design a predictive model to identify exact attack types

c) Design and utilize a visual analytic tool to represent their distinctive patterns and support an interactive visual analysis on the patterns

d) Understanding cloud-based network intrusion detection techniques

4 BACKGROUND & MOTIVATIONS

4.1 Intrusion detection

Due to the advancement of Internet technologies, applications, and protocols, network traffic analysis has become a complicated field since it deals with an extreme amount of network traffic data. Because of the network complexity, network traffic analysis to detect unauthorized network intruders is also considered as one of the increasingly important research topics in network security. Kemmerer and Vigna [1] outlined a handy history of intrusion detection, moving from the early days of manual detection and analysis by systems administrators in the 1970s to real-time solutions in the 1990s and early 2000s. Intrusion detection is defined as the process of monitoring events that occur on computers and networks ([2]). As intrusions became more varied and sophisticated, the need for frameworks and classifications has been emphasized. Debar et al. [3] performed a taxonomical approach of understanding intrusion detection system by defining them as mechanisms that process

information coming from the system that is to be protected. They further described existing intrusion detection techniques using two important concepts: detection method and behavior on detection.

Intrusion detection methods are often classified as signature-based and behavior-based or anomaly-based techniques [4]. Signature-based techniques (also known as knowledge-based or misused-based techniques) reference databases of previous attack signatures and known system vulnerabilities. Signature-based techniques are quite accurate and effective for known attacks, but cannot guard against unknown attacks. To reduce this limitation, constant update of attack signatures needs to be performed. However, this might require considerable resources and overhead. To guard against unknown attacks, behavior-based or anomaly-based techniques can be used. These techniques detect intrusion attempts as deviation by comparing them to normal network activity (i.e. commands or traffic). While the two types of techniques maintain considerable overlap and are often regarded as identical in literature, we posit that they are slightly different. Hereafter, we will refer to anomaly-based techniques as those that involve first training a system to establish a normal profile, and then use the profile to detect deviations, and behavior-based techniques as those that do not necessarily compare against a baseline. For example, in behavior-based detection methods, an administrator might simply establish certain rules that would trigger alerts when broken. In practice, the two types of techniques are often the same, but it is nonetheless important to establish their subtle differences.

Once effective detection methods have been established, the question becomes: what kind of behavior does the system adapt after detection? While Debar et al. [3], for example, mentioned behavior on detection, we prefer behavior "around" detection, as it better describes a system's possible actions both before and after detection - intrusion detection system is hardly ever simply reactionary systems that only take action after the fact. Along these lines, Halme and Bauer [5] advanced one of the first taxonomies of anti-intrusion techniques and divided them into six approaches: prevention, preemption, deterrence, deflection, detection, and countermeasures. The first three approaches (prevention, preemption, and deterrence) are passive measures to guard against attacks, and the latter three (deflection, detection, and countermeasures) are active measures to protect elements in a system. Many of these approaches are fluid and can be used at many points throughout the process - before, after, or during attacks. Regardless of the terminology or order of deployment, intrusion detection system is considered as a critical system that detects and acts against attacks in a variety of ways. In the late 1990s, researchers [3, 6] inventoried early real-time intrusion detection systems, many of which employed combinations of signature and behavior-based techniques by following Halme and Bauer's six approaches [5]. As time wore on, however, and network traffic continued its inexorable growth, these systems became untenable in real-time, so researchers have taken to the cloud to beef up analysis and computing of network intrusion detection.

The Internet is a globally distributed network that supports communications between various software applications and computer systems, which generate a numerous amount of different network traffic patterns [7]. With the network traffic, it is possible to perform an analysis of identifying the usages of network resources because traditionally known applications (e.g. email, WWW, or

else) or new internet-based applications (e.g. online games, peer-to-peer (P2P) file sharing, and among others) generate most network traffic data. Network administrators monitor network traffic to identify a possible network congestion. If needed, a reallocation of network resources is performed to guarantee a reliable network communication. Therefore, it is possible for us to communicate or share data seamlessly through the Internet without delay. Since unusual network activity may slow the communication speed down, a study of identifying anomalous network traffic or behaviors has been spotlighted in the network security community. In network monitoring, accurate and rapid abnormal internet traffic detection is critical [8]. Thus, anomalous network traffic or behaviors should be filtered out to guarantee a smooth network communication. Identification of the applications responsible for generating Internet traffic is commonly performed by locating well-known service ports obtainable from network packet header [7]. Since numerous emerging applications and services do not use well-known ports, the technique of employing known ports (e.g. 80, 22, or else) to create a tunnel to other applications is broadly adopted. Therefore, analyzing internet traffic based on known port numbers is no longer an effective approach. More specifically, a port-based classification is ineffective for identifying the usage of P2P applications. With the port-based classification, 30-70% of internet traffic is classified as "unknown" [9]. Instead of using known ports, many current applications use dynamic ports. Due to the limitation of identifying network flows using service ports, researcher designed new methods by considering application payload signatures as a deep packet inspection method [10] and a payload-based method [11]. These methods directly compare stored signatures to the packets coming from applications. Since new applications are emerging and existing application protocols keep upgrading, the methods have a limitation of analyzing future internet traffic. Due to this limitation, a flow-based classification received much attention [12, 13]. This approach performs a classification based on various flow features such as the number and size distributions of internet packets in a network flow, flow duration, and inter-packet arrival time [7, 14]. To overcome the shortcomings of the approaches that use port and signature information, researchers started using statistical methods to classify internet traffic flows. They mainly focused on identifying statistically valid characteristics from the network traffic flow [7, 14, 15, 13, 16, 17, 18, 19, 20]. For instance, Moore and Papagiannaki [13] generated more than 200 features from the Internet traffic data. Later, these features have been broadly used to perform extensive studies on identifying best analytical approaches [12, 15, 7].

4.2 Techniques for intrusion detection analysis

To address the issue of protecting computing infrastructures, researchers have proposed numerous intrusion detection (ID) techniques. A traditionally known ID technique uses a network intrusion detection system to discover threats by analyzing network traffic at the network layer. The intrusion detection system (called host-based IDS) identifies threats on computer hosts by monitoring computer system logs, system calls, network events, and files [21]. In general, network intrusion detection system detects abnormality by identifying intruders' threat signatures. To detect any abnormal

behaviors, it monitors network packets to find possible attack signatures and compare them to known attacks patterns. Since the system detects threats based on known attacks' signatures, new attacks cannot be detected [22]. Although the host-based IDS is designed to prevent intruders from changing computer system security policies, it cannot monitor network traffic precisely because it only can detect intrusions based on the analysis of the information such as logs or packets [23]. Most analysis approaches are designed to detect intrusions by conducting misuse detection and anomaly detection. The misuse detection is looking for events (i.e. known attacks) that are matched to predefined signatures [24, 25]. The anomaly detection identifies abnormal behaviors on hosts or networks based on the assumption that each attack shows somewhat different behaviors compared to normal activity. Therefore, it is possible to identify any abnormal attacks without specific knowledge by finding their patterns. So, anomaly detection is also applied to design various applications in other areas such as credit cards fraud detection [26], fault detection in safety critical systems [27], and any domains that aim to detect abnormal activities including a medical field [28]. For example, an abnormal pattern of CT images from a patient may discover unexpected diseases. Detecting unusual credit card transactions can be used to detect thieves [29]. Although this method can identify abnormal behaviors without having any knowledge, it provides a high false alarm rate and requires extensive training sets to get a reliable performance result [30, 31].

In the past, researchers studied on increasing the rate of detecting network attacks. Cheng et al. [32] proposed an approach of identifying normal TCP flows by using spectral analysis techniques to protect legitimate TCP flow from Denial of Service (DoS) attacks. Wang et al. [33] proposed a statistics-based approach to detect TCP SYN flood attacks, which uses a nonparametric cumulative sum (CUSUM) method. Utilization of statistical approaches is good for maintaining high accuracy with spending reasonably short detection times because it approximately calculates normal traffic patterns to perform a comparison with abnormal traffics. However, it has a difficulty of detecting anomalies caused by network system failures. To resolve this difficulty, Thottan and Ji [34] used a statistical data analysis method with a signal processing technique together to quantify network behaviors to understand network anomalies. They classified network anomalies into two categories as network performance anomalies (e.g. file server failures, paging across the network, broadcast storms) and security-related problems or attacks. They showed that their approach of integrating a signal processing technique is effective for detecting several network anomalies. However, an accurate statistical model was not utilized to detect different abnormal traffic patterns. Artificial neural [35] has been applied broadly because it has a potential of identifying and classifying unknown network activities [36]. Lippmann and Cunningham [37] utilized neural networks to design a detection model by searching for attack-specific keywords in network traffic. Sarasamma et al. [38] used multilevel hierarchical Kohonen Net (K-Map) consisting of three layers to determine different types of attacks. To increase the speed of selecting features, input dataset is divided into three feature sets based on domain knowledge. After applying single-layer K-Map onto each feature set, significant subset features are determined and used to design next hierarchical K-Map. Hand and Cho [39] proposed

an approach of employing an evolutionary neural network (ENN) to overcome the limitation of designing a correct topology (i.e. domain-specific neural network model) for detecting network attacks. Rule-based anomaly detection techniques are introduced to capture rules that can identify network behaviors using Fuzzy [40, 41] or decision trees [42, 43, 44, 45]. Also, clustering technique [46] and SVM [47, 48, 49, 50] are used by numerous researchers to detect abnormal network behaviors. Since SVM supports both supervised and unsupervised learning, Shon and Moon [51] applied a hybrid approach of integrating the two learning methods with emphasizing the advantages of utilizing both SVM approaches. Similarly, Jain and Abouzakhar [45] designed an approach by utilizing both Hidden Markov Model (HMM) and Support Vector Machine (SVM). Xiang et al. [52] introduced a multiple-level hybrid classifier combining tree classifiers and Bayesian clustering to detect network anomaly, Kuang et al. [47] presented a hybrid classifier by integrating SVM and principal component analysis. Golmah [53] showed a hybrid intrusion detection method with integrating both C5.0 and SVM.

To generate a reliable ID system model, feature selection and feature extraction are considered as critical tasks for saving computational cost as well as for discovering data patterns. The feature selection is used to select a subset of most meaningful features from the original feature. The feature extraction is necessary for converting input data to reduce dimensions. There are various techniques that can be used for the feature extraction and feature selection such as Genetic Algorithm (GA) [49], entropy measure of network features [54], Partial Least Square (PLS) [55], Kernel Principal Component Analysis (KPCA) [47], and cuttlefish optimization algorithm [56]. When applying the feature extraction, there is an important consideration whether the characteristics of original input data are transmitted to extracted new feature sets. However, it is important to note that the generated new feature set may not maintain the same or similar patterns compared to the original input data [57]. Sanei et al. [58] addressed the potential capability of discovering important features from input data by utilizing signal processing techniques.

4.3 Cloud-based network intrusion detection

As traffic grew and attacks became more prevalent with the popularity of the Internet in the 1990s, it has become clear that swifter intrusion detection analysis was necessary to both diagnose and prevent attacks. To accomplish this, researchers worked to understand network traffic patterns, which resulted in the greater development of signature-based and behavior-based detection techniques. While both techniques can be effective in real-time, significant limitations nonetheless exist - signature-based techniques cannot guard against unknown intrusions, and behavior-based techniques break down under heavy traffic or sudden traffic bursts [25].

To address these heavy traffic limitations, researchers harnessed the power of cloud computing technology to speed up computation. In recent years, the MapReduce computing platform, particularly through the Hadoop distributed file system, has been used to perform advanced intrusion detection analysis [59, 60, 61, 62, 63]. Hadoop, a popular open-source software framework for distributed storage and distributed processing of big data, uses MapReduce, a parallel processing

paradigm that can perform rapid analysis to determine the presence of attacks or malicious activities in large quantities of network traffic. Previous studies have shown the effectiveness of using cloud computing technology for intrusion detection analysis [64, 65, 66], but recent literature [67] emphasizes the importance of machine learning algorithms (ML) to intrusion detection analysis. Although the aforementioned approaches and techniques significantly improved intrusion detection techniques throughout past decades, many researchers have argued for the importance of ML in intrusion detection.

ML cannot be applied to intrusion detection directly. The integration of ML to intrusion detection has not been studied broadly, primarily due to the problems associated with leveraging optimal algorithms in a cloud computing environment. Sommer and Paxson [67] outlined challenges for ML in intrusion detection, highlighting the fact that machine learning tools are most adept at finding activity similar to something previously seen, which goes against the general definition of the anomaly-based intrusion detection techniques that seek to identify novel attacks. Also, the high cost of errors, lack of training data, and enormous variability of input data all impede the applicability of ML to intrusion detection. Despite these considerable challenges, researchers have tackled problems and developed ML for intrusion detection. Below, we will discuss recent advancements in the field of ML in intrusion detection, highlighting the research that is successfully addressing the challenges that Sommer and Paxson [67] first set forth in 2010. This research of understanding cloud-based network intrusion detection employs two types of ML - clustering and classification algorithms - which are considered suitable for intrusion detection techniques.

4.3.1 Clustering algorithm

Clustering is a form of unsupervised machine learning that does not rely on training data or classification models. Instead, it splits input datasets into clusters by common features to find similar patterns in input datasets. Similarity measures (e.g. Euclidean distance) are often utilized to uncover these patterns. The *k*-means algorithm is a simple, clustering algorithm popular for general use [68]. The algorithm works by first selecting initial cluster centers (also known as centroids) and calculating the average distance between centroids and all other points in the system. This step can then be iterated continuously, establishing new centroids and relocating data points until the average distance is decreased and no more relocation occurs. Clustering algorithms can be used in a variety of fields, including market segmentation, geostatistics, computer vision, search, and medicine, among others. They are also widely used as preprocessing steps for other algorithms, where they can be useful in providing initial configurations. More importantly for our purposes, however, clustering algorithms have been used to tackle the important problem of network traffic classification. In the past, traffic classification was accomplished with entirely port-based and payload-based techniques. However, with the advent of applications that routinely use dynamic port numbers, masquerading techniques, and encryption, clustering algorithms have been implemented to exploit applications' distinctive characteristics in behavior-based approaches.

Clustering algorithms explored in recent network traffic research include the *k*-means, DBSCAN, Expectation Maximization, and AutoClass algorithms. Nguyen and Armitage [69] concluded that the AutoClass algorithm produces the best accuracy when performing clustering for network traffic classification. McGregor et al. [70] found Expectation Maximization to be useful in finding network flow statistics. Bernaille et al. [71] employed *k*-means to classify traffic via TCP headers, possibly eliminating the need to collect whole packets to predict intrusion. From these researches, it is quite evident that clustering algorithms provide salient applications to network traffic analysis, and to intrusion detection on the whole. Computing platforms such as Hadoop and MapReduce distribute computations among scores of computers or more, stymieing any algorithms with iterative or linear computations that require access to all input data to perform. Many popular ML cannot be directly utilized in cloud server architecture for this reason [72].

4.3.2 Classification algorithms

Unlike clustering, classification algorithms provide supervised machine learning for network intrusion detection. In this sense, good practice suggests that the presence of large learning datasets lead to higher probability of success. Classification consists of two stages: training and testing. In the training step, a classifier model is selected to receive learning input, and once the classifier model has been sufficiently trained, testing is performed to determine accuracy through false positives, false negatives, true positives, and true negatives. A good classifier model, as we expect, returns a minimal quantity of false positives and negatives.

In computer networks, classification algorithms can be used to perform categorization of packets into "flows," in a process known as packet classification [73]. Packet classification has been used to support access control, quality of service, and intrusion detection [74]. Broadly used classification algorithms include Naïve Bayes (NB), Bayesian Network (BN), Logistic Regression (LR), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Tree (DT), Random Tree Classifier (rTree), Genetic Algorithms (GA), and Random Forest Classifier (rForest) [69, 75, 76, 77, 78, 79].

In the past, statistical analysis has been used broadly to detect intrusions by performing a statistical comparison of current network events to a pre-determined set of baseline criteria. Since the statistical analysis has a limitation of identifying different types of attacks, researchers proposed various alternative approaches. Wang [80] showed the effectiveness of using logistic regression modeling to detect multi-attack types. Cannady [81] emphasized the usefulness of using ANN for intrusion detection. However, ANN has not been widely applied since the accuracy of detecting intrusions is closely depending on used training datasets and methods. Also, it does not provide a detailed level of accuracy (including reasons). Therefore, it has been known as "black box" operation. NB is known as a simplified Bayesian probability model. NB classifier operates based on the likelihood that one attribute does not affect others. Amor et al. [82] utilized NB for detecting intrusions. Although NB is faster than DT for learning and classifying, they identified that there was no significant performance difference between the two for detecting intrusions. rForest is also one of the broadly used classi-

fication algorithms in intrusion detection. In recent, Albayati and Issac [83] found that rForest is more effective than NB for detecting intrusions. From the study of measuring the performance of detecting intrusions among NB, rTree, and rForest, they identified that rForest is superior to others with maintaining low false alarm rate. Interestingly, many researchers used SVM to conduct intrusion detection analysis [84, 85, 86, 87, 88] because it is good for classifying data by finding the hyperplane that maximizes the margin among all intrusion classes. It simply classifies the input data by a set of support vectors representing data patterns. However, SVM classification depends mainly on used kernel types and parameter settings [88]. It also requires longer training time than other classification algorithms. To address this limitation, Khan et al. [84] proposed an approach of integrating hierarchical clustering analysis. Different from other classification algorithms, GA approach has been in used for various purposes in intrusion detection as optimization, automatic model generation, and classification [89]. GA is a search algorithm that utilizes the mechanics of natural selection and genetics. It is often used to generate detection rules or to select appropriate features from the input data. However, the classification accuracy of using GA was slightly lower than tree algorithms such as J4.8 and CART [90]. Although numerous classification algorithms are used to detect intrusions, only a couple of algorithms have been used with an integration of cloud computing architecture.

4.3.3 Clustering Vs. Classification

As we discussed above, in the past, researchers focused on identifying different algorithms' applicability to intrusion detection. However, in recent studies [78, 77], researchers has exclusively focused on the use of classification MLAs in network intrusion detection. From the study by Erman et al. [75], it has been found that clustering (i.e. AutoClass) outperforms classification (i.e. Naïve Bayes) by up to 9% in accuracy metrics like recall, precision, and overall accuracy.

Why then, do classification algorithms dominate the current literature on MLAs in intrusion detection analysis? Although clustering can produce better results than classification, it is important to note that clusters do not map 1:1 to applications [75]. In an ideal clustering environment, the number of clusters would equal the number of application classes (HTTP, SMTP, FTP, POP3, etc.) and each application class would dominate one cluster [69]. In reality, applications can spread out and dominate many clusters or dominate no clusters at all. In these situations, it can become quite sticky to map backward from a cluster to the source application. Given this limitation, classification algorithms are much more commonly applied in the network intrusion detection sphere.

By and large, these MLA experiments and implementations relied on static, offline analysis of previously captured traffic. As network traffic data grows and more of these MLAs are adapted to the cloud, utilization of cloud computing for network intrusion detection is increasingly inevitable. In the following section, we discuss recent advances in the field.

4.3.4 Identifying network abnormal behaviors using different distance measures with MapReduce

In recent decades, the Internet communication has become broadly used in business, education and learning, and entertainment. On the Internet, an enormous amount of data is generated by people. Since personal or business information can be interrupted by intrusive activities on the Internet, the field of Cyber-Security has grown and deploying many methods to prevent or stop cyber attackers as firewalls, anti-virus software, and intrusion detection systems. From those methods, intrusion detection system (IDS) is one of the most suitable methods over the Internet. IDS can provide an effective defense to the information stored in the network systems. A primary objective of deploying the IDS is to recognize abuse, illegitimate use, and misuse of network system attacks and to prevent them from carrying out their attacks. Most recent intrusion detection methods have focused on utilizing machine learning techniques for automating the detection process [91]. This technique utilizes k-means algorithm to categorize network traffic data into clusters. Since a utilization of clustering technique with cloud computing techniques has become a suitable method to detect anomalous activities from the network traffic data, our study focuses on utilizing MapReduce technique to handle the huge amount network data (i.e. NSL-KDD dataset). MapReduce runs under the open source framework Hadoop, which provides parallelization, fault-tolerance, data distribution, distributed processing, automatic parallelization, large-scale distributed computing and a simple interface it is ideal for running k-means Clustering [92]. More specifically, this study focuses on identifying best suitable distance measures to detect network intrusions by performing k-means clustering.

4.4 Visual Analytics to detect network intrusion

In the early network intrusion detection literature, statistical analysis was broadly applied to detect intrusions by performing a statistical comparison of current network events to a pre-determined set of baseline criteria. However, the statistical analysis has a limitation of identifying new types of attacks. Since the popularity of the Internet produces a massive amount of network traffic patterns as well as unknown attacks, designing an innovative intrusion detection analysis is considered as a major research area in cybersecurity. To accomplish this challenge, researchers proposed two intrusion detection techniques: signature-based detection techniques and behavior-based or anomaly-based detection techniques [4].

Signature-based techniques (also known as knowledge-based or misused-based techniques) are based on known attack signatures or system vulnerabilities. Since signature-based techniques analyze current network traffic patterns by comparing them with the attack signatures, they are quite accurate for detecting known attacks, but less effective against unknown attacks. Therefore, a constant update on attack signatures should be performed which may require considerable resources and overhead. Behavior-based or anomaly-based techniques perform an analysis of detecting intrusions via deviations from normal or expected traffic behaviors. Although the two types (anomaly vs. behavior)

maintain considerable overlap and are often regarded as identical in literature, it is important to note that there is a slight difference between them. Anomaly-based techniques create a normal profile by training current network traffic, and then use the profile to detect deviations. On the other hand, behavior-based techniques do not necessarily compare against a baseline profile. Although these techniques can be effective to detect network intrusions in real-time, there are significant limitations. Signature-based techniques cannot guarantee to detect unknown network intrusions, and behavior (or anomaly)-based techniques show a significant performance issue under heavy traffic or sudden traffic bursts [25]. To address the limitations, researchers have studied on designing new techniques which integrate different Machine Learning (ML) techniques [78, 77].

In 1990's, Cannady [81] emphasized the usefulness of Artificial Neural Networks (ANNs) for intrusion detection. However, ANNs are not used widely since the accuracy to detect intrusions is closely dependent on datasets and methods used in training. Furthermore, they do not provide a detailed reason about detected intrusions. Amor et al. [82] utilized Naïve Bayes (NB) for detecting intrusions. As a simplified Bayesian probability model, NB classifier operates based on the likelihood that one attribute does not affect others. It is faster than Decision Tree (DT) for learning and classifying, but there is no significant performance difference between the two. As the number of studies on designing new intrusion detection techniques by adapting ML algorithms increased in early 2000, Nguyen and Armitage [69] surveyed various network traffic classification algorithms appeared in the period between 2004 and early 2007. They found that different ML algorithms such as AutoClass, Expectation Maximisation (EM), DT, and NB demonstrate high accuracy. However, most approaches are uniquely designed to define their classification models by evaluating different test datasets. Therefore, the models are less efficient to analyze different datasets and network circumstances. To overcome this limitation, researchers have continuously sought and adopted various new ML algorithms. Wang [80] showed the effectiveness of logistic regression (LR) modeling to detect multi-attack types. Albayati and Issac [83] compared the performance of detecting intrusions among NB, Random Tree Classifier (rTree), and Random Forest Classifier (rForest), and found that rForest is superior to others with maintaining low false alarm rate.

Many researchers utilized Support Vector Machine (SVM) to conduct intrusion detection analysis. SVM is well-suited for classifying data by finding the hyperplane that maximizes the margin among all intrusion classes [84, 85, 86, 87, 88]. It simply classifies the input data by using a set of support vectors representing data patterns. However, SVM classification depends mainly on the used kernel types and parameter settings [88]. It also requires longer training time than other classification algorithms. To address this limitation, Khan et al. [84] proposed an approach of integrating hierarchical clustering analysis. Genetic Algorithm (GA) has been used for various purposes in intrusion detection such as optimization, automatic model generation, and classification [79, 89]. GA is a search algorithm that utilizes the mechanics of natural selection and genetics. It is often used to generate detection rules or select appropriate features from the input data. However, the classification accuracy using GA is slightly lower than tree algorithms such as J4.8 and Classification and Regression Trees (CART) [90].

CART is an algorithm that generates a set of rules by splitting data into each child node. Since CART predicts continuous dependent variables (regression) and categorical predictor variables (classification) by building a tree, it is used to perform a classification [93, 94]. It also supports dealing with multiple data types and missing values. Despite the fact that classical PCA has high sensitivity to outliers [95], PCA is often used to extract significant features from network traffic data. In summary, numerous studies have been conducted to design an effective network intrusion detection technique. However, it is important to note that one algorithm or approach cannot detect all existing or unknown attacks precisely due to the existence of anonymity in network traffic patterns.

In the visualization community, researchers apply visualization techniques to address limitations in traditional network intrusion detection analysis. Due to the importance of analyzing large complex network traffic data, the community provides network traffic data as a part of visualization challenge to motivate people to analyze real-world network traffic data visually. For example, the VAST 2012 challenge [96] provided a financial institution's data for researchers to gain an opportunity to understand the health of global corporate network visually by identifying anomalies or problems. Shiravi et al. [97] conducted a comprehensive review of existing network security visualization systems by classifying them into different five use-case classes: host/server monitoring, internal/external monitoring, port activity, attack patterns, and routing behavior. Although numerous network security visualization systems have been designed, most systems focus only on addressing the issue of how to represent collected log data or network events. To better understand network traffic patterns and detect network intrusions more precisely, visualization techniques should be integrated with computational and machine learning approaches. In the following sections, a detailed explanation how we combine both computational approaches and visualization techniques in intrusion detection is included.

5 METHODOLOGY

This section includes a detailed explanation how to generate a multi-level network abnormal detection. Figure 1 represents the overall procedure of the proposed prediction model.

5.1 Dataset

This project used two datasets. A dataset, a publicly available internet traffic dataset [98], is used for testing our proposed wavelet features. A dataset, a publicly available intrusion detection dataset (called NSL-KDD dataset [99, 100]), is used to design a predictive model based on the wavelet features for our final goal.

5.1.1 Internet traffic dataset

The dataset is generated from both network link directions on Genome campus with three institutions on the site. A detailed explanation about this dataset can be found in [98, 13]. Since the dataset

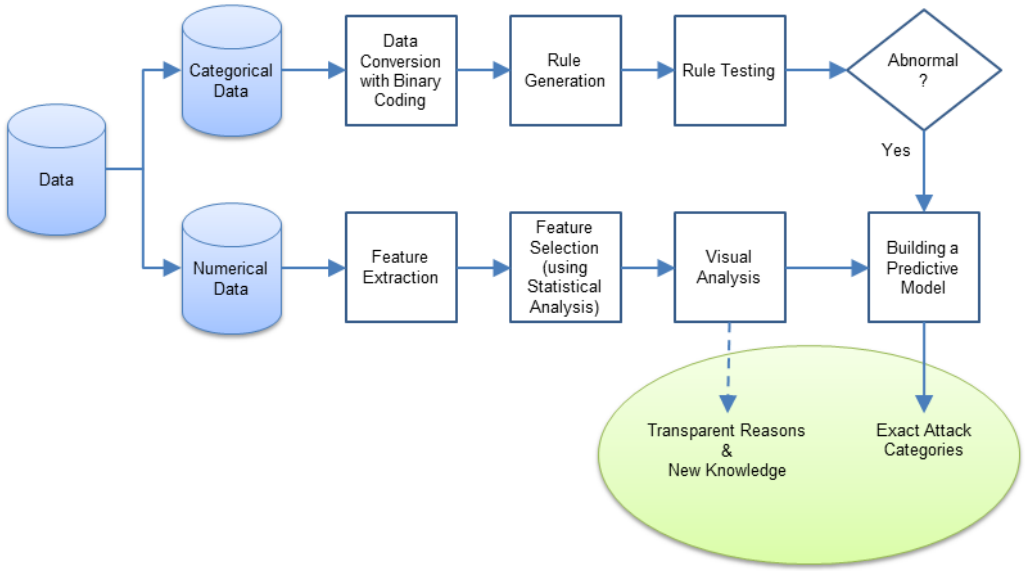


Figure 1: A schematic diagram of the proposed approach

provides classification information about packets, researchers used this dataset to design new approaches to detect abnormal behaviors [7, 15, 12]. The data used in this study includes computed mean, median, maximum, minimum, and a variance of packet inter-arrival times. Throughout this project, we call this data "raw features."

Table 1: A description of used normal and abnormal datasets. Three classes (i.e. P2P, Mail, FTP) represent normal behaviors and one class (i.e. attack) indicates abnormal behavior. Each number in this table indicates the number of samples (i.e. data records)

	Class	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5
Normal	FTP	210	210	209	209	210
	Mail	209	209	210	210	209
	P2P	209	209	209	209	209
	Total	628	628	628	628	628
Abnormal	Attack	628	628	628	628	628

Most previous studies commonly used imbalanced internet traffic data when designing a predictive model to detect abnormal behaviors. If a predictive model is generated with imbalanced datasets, it can detect a majority class while maintaining good accuracy. However, it has been known that it is difficult to detect a minority class due to the influence of the large majority class [101]. Specifically, it cannot classify a new incoming minority class correctly. Researchers found that the performance of existing classifiers tends to be biased towards the majority class because of unequal class distribution [102]. To overcome this limitation, balanced dataset is used to avoid possible bias caused by the majority class. In the dataset, three classes (P2P, Mail, FTP) are classified as "normal" behavior and the class (attack - virus and worm attacks) is considered as "abnormal" behavior. The original dataset is an imbalanced dataset. That is, the normal behaviors are considered as the majority class and the

abnormal behavior is regarded as the minority class. To balance the normal and abnormal data, the same sample size of the normal and abnormal data is used for this study. The abnormal (i.e. attack) data contains m number of samples. To balance the normal and abnormal data, the same sample size of the normal data is used. That is, a total N number of normal samples are divided into k disjoint datasets as $n_1, n_2, n_3, n_4, \ldots n_k$ so that $n_i \in N, i = 1,2,3 \ldots k$ and $n_i \cap n_j = \phi$. Each n_i dataset includes the same numbers of abnormal data. Thus, the balanced datasets $d_i (i = 1,2,\ldots,k)$ are generated by combining both normal and abnormal datasets (see Table 1). In our study, six packet inter-arrival time (IAT) information (see Table 2) is used to address the utilization of our proposed approach integrating both the signal processing technique and logistic regression (LR) for internet traffic monitoring.

Table 2: Six inter-arrival time (IAT) features for all packets (considering both directions) are used in this study.

Feature name	Description
(f1) min IAT	Minimum packet inter-arrival time
(f2) q1 IAT	First quartile inter-arrival time
(f3) med IAT	Median inter-arrival time
(f4) mean IAT	Mean inter-arrival time
(f5) max IAT	Maximum packet inter-arrival time
(f6) var IAT	Variance in packet inter-arrival time

5.1.2 NSL-KDD dataset

In this study, a publicly available intrusion detection dataset (called NSL-KDD dataset [99, 100]) is used. NSL-KDD dataset is the refined version of the KDD cup'99 dataset [100]. Since the KDD Cup'99 dataset includes many redundant records, inefficient learning process and unreliable accuracy are caused. To resolve this issue, repeated records are removed and named as NSL-KDD dataset [99]. The NSL-KDD dataset includes a training set (125,973 records) and a testing set (22,544 records). It contains 41 attributes (three nominal, six binary, and thirty-two numeric attributes), and includes normal activity and twenty-four attacks. These attacks are grouped into four major categories. In this study, the training and testing data are combined to make a new input data. That is, total 148,517 data is used as an input data. Table 3 presents the four major attacks and its intrusion types.

Table 3: Four attack categories in the NSL-KDD dataset.

Four categories	Intrusion types
DoS	back, land, neptune, pod, smurf,teardrop, mailbomb,processtable
R2L	ftp_write, imap, guess_passwd, multihop, phf, spy, warezclient, warezmaster sendmail,snmpgetattack,snmpguess,worm,xlock,xsnoop,named
U2R	buffer_overflow, loadmodule, perl, spy, rootkit,ps, xterm, sqlattack,mscan
Probe	ipsweep, nmap, portsweep, satan, saint

DoS attack indicates attempts to disabling network access to remote machines (or computing resources). R2L represents that a remote user gains access to local user accounts by sending packets to a computing machine over the network. Probe explains that network is scanned to gather information

to find known vulnerabilities.

5.2 Verification of Wavelet features using imbalanced dataset

The Internet is a globally distributed network that supports communications among various applications and computer systems that generate different network traffic patterns [7]. With the analysis of the network traffic patterns, we are able to identify the usage of network resources. Network administrators monitor network traffic to identify possible network congestion. If needed, reallocation of network resources is performed to guarantee reliable network communication. Therefore, we are able to communicate or share data seamlessly through the Internet. Since unusual activity may slow the network communication speed down, a study of identifying anomalous network traffic or behaviors has been regarded as one of the important researches in the network security community. In network monitoring, accurate and rapid abnormal internet traffic detection is critical [8]. Thus, anomalous network traffic or behaviors should be filtered out to guarantee smooth network communication.

Identification of the applications responsible for generating internet traffic is commonly performed by locating well-known service ports obtainable from network packet header [7]. Since numerous emerging applications and services do not use well-known ports, the technique of employing known ports (e.g. 80, 22, or else.) to create a tunnel to other applications is broadly adopted. Therefore, analyzing internet traffic based on known port numbers is no longer an effective approach. More specifically, a port-based classification is ineffective for identifying the usage of P2P applications. With the port-based classification, 30-70% of internet traffic is classified as “unknown” [9]. Instead of using known ports, many current applications use dynamic ports. Due to the limitation of identifying network flows using service ports, researcher designed new methods by considering application payload signatures as a deep packet inspection method [10] and a payload-based method [11]. These methods directly compare stored signatures to the packets coming from applications. Since new applications are emerging and existing application protocols keep upgrading, the methods have a limitation of analyzing future internet traffic. Due to this limitation, a flow-based classification receives much attention [12, 13]. This approach performs a classification based on various flow features such as the number and size distributions of internet packets in a network flow, flow duration, and inter-packet arrival time [7, 14]. To overcome the shortcomings of the approaches that use port and signature information, researchers started using statistical methods to classify internet traffic flows. They mainly focused on identifying statistically valid characteristics from the traffic flows [7, 14, 15, 13, 16, 17, 18, 19, 20]. For instance, Moore and Papagiannaki [13] generated more than 200 features from the Internet traffic data. Later, these features have been broadly used to perform extensive studies on identifying the best analytical approaches [12, 15, 7].

Imbalanced data often occurs in various domains including medical, biology, and computer networks [103]. In the network security community, network features or best combinations of the features from imbalanced internet traffic data are used to identify abnormal behaviors. If a predictive model is designed with the imbalanced data, it cannot classify minority class successfully because the

model determines all new incoming data as majority class [101]. In addition, most previous studies mainly focused on analyzing the internet traffic data by utilizing their actual values (i.e. raw data) as input features. Therefore, there might be a limitation of detecting any sudden changes within the data as abnormal traffic behavior.

5.3 Creating reliable rules to detect network abnormality

5.3.1 Pre-Processing

As mentioned above, the NSL-KDD dataset contains three nominal variables that include protocol type, service, and flag. However, each nominal variables contains many distinctive attribute values. Protocol type includes three attributes (i.e. TCP, UDP, and ICMP), service includes 70 attribute values (i.e. SMTP, HTTP, POP3, SSH, WHOIS, and among others), and flag contains 11 attributes (i.e. SF, S2, S1, S3, REJ, RSTR, and among others). Since the nominal variables contain numerous amount of attribute values, it is difficult to extract transparent information regarding network abnormality. To resolve this issue, a binary coding scheme [104] via the use of indicator variables is applied to the three nominal variables. Binary coding uses 1 (“one”) to indicate the occurrence of a category of interest and 0 (“zero”) to represents its nonoccurrence [105]. For example, if the attribute value of protocol type is “TCP”, it is converted to 1, and otherwise 0.

When labeling all attacks as “abnormal”, total of 77,054 normal and 71,463 abnormal data are formed. To generate a rule-based method to identify abnormal behaviors, nominal and binary variables are used. By reforming the nominal variables, total of 90 features including binary variables (i.e. yes/no) are generated. Since the binary coding to the nominal variables causes an increase of data dimensions, important features are selected. For this selection, a statistical validation using SAS is performed. Then, each normal and abnormal data are randomly divided into 10 different subsets to apply ten-fold cross validation.

5.3.2 Rule generation with CART

To design a rule-based model, Classification and Regression Tree (CART) [106] is used. CART applies the concept of information theory to create a decision tree that captures complex patterns of input data. It is broadly used due to its efficiency in dealing with multiple data types and missing values. CART expression forms explicit and transparent grammatical rules [107, 108]. Thus, it is much simpler to understand data patterns than other models. Also, it uses an exhaustive search of all variables and split values to find optimal splits for each node by measuring the degree of impurity for each outcome of the feature. To find the most important features for identifying network traffic abnormal behaviors, a statistical test (i.e. ANOVA) is applied. Then, decision trees are generated from each training set using the selected significant features. Due to the difficulty of extracting rules from the generated trees, a software application (called *TreeParser*) is designed to extract rules from the trees by navigating all branches of the generated trees. With the extracted rules, the performance of each rule is measured

with a distinctive testing dataset.

5.4 Exact attacks detection

Whenever incoming network traffic events are identified as "abnormal behaviors," it is important to determine exact types of the behaviors (i.e. attack types/ categories). Providing exact information is important for system administrators to protect computing infrastructures by initiating relevant actions. In this study, three attack categories (i.e. DoS, Probe, R2L) are used due to the lack of U2R data.

5.4.1 Feature extraction using signal processing technique

Wavelet analysis is also widely used for analyzing abnormal events because of its ability of identifying hidden patterns from time-frequency information by separating input data into different levels of frequencies. Applying the signal processing technique (i.e. wavelet analysis) to data helps us isolate the characteristics of the traffic by extracting hidden patterns of high and low frequency information. Most previous studies that utilized wavelet transform techniques focused on detecting abnormal patterns by performing data reconstruction, two or three frequency range analysis, and filtration (or thresholding). Since one of the best advantages of applying wavelet analysis is the possibility of extracting information at different levels of signal decomposition (i.e. frequencies), extracting valuable features to discover underlying patterns is a crucial step for identifying network abnormal behaviors. However, it has been known that existing network anomaly detection methods based on wavelet transformations have a limitation of using low frequency anomaly [109]. In this study, we address this important consideration when designing a predictive network abnormal behavior model. Table 4, various techniques (as major steps) were used to detect network anomalies. Although threshold techniques or coefficient measures were commonly applied to determine input features, our approach utilizes a statistical measure to identify statistically significant features and use them to design a predictive model.

Since signal processing technique has a capability of discovering hidden patterns from input data, discrete wavelet transform (DWT) is used. DWT is a broadly known promising method for time-frequency analysis. Since wavelet indicates a small localized wave in a time domain, any sudden or rapid changes in the data can be identified easily. Because of the characteristics that DWT has, it is advantageous to analyze non-stationary signal data (e.g. network traffic data) with identifying important patterns. DWT decomposes the input data into different levels of frequency component by calculating its correlation with a set of chosen wavelet basis function [110, 119, 120]. The ability to preserve both time and frequency resolutions has led to widespread use of DWT in various practical application domains [101]. It is particularly good for local analysis in representing fast time-varying and non-stationary signals like network traffic data. The merits of using DWT are (a) analyzing non-stationary time series data (e.g. internet traffic data), (b) capturing the non-stationary nature of the data in the time-frequency domain, (c) detecting any rapid changes in the data, and (d) revealing

Table 4: A summary of researches utilizing wavelet transformations are presented‘[].

Work	Goal	Dataset(s)	Used features	Major step(s)	Used algorithm(s)
[110]	Network anomaly detection	(1) Simulated data (2) BTnet Dial IP service data	(1) Simulated data features and (2) Set of network metrics	Estimation of Nuisance parameters by maximum likelihood using wavelet coefficients	Undecimated DWT and Bayesian
[111]	Network anomaly detection	CAIDA dataset on the Witty Worm	Not known	Two thresholds for the compression and event detection tasks	Haar wavelet, adaptive threshold, and Donoho - Johnstone universal threshold (aka VisuShrink)
[112]	Network anomaly detection	Simulated data	Decomposed three distinct signals (low/mid/high) from SNMP and IP flow dataset	Deviation scores from three range (low, mid, and high) frequency information	Wavelet analysis and Time frequency-localization
[113]	Network anomaly detection	(1) DARPA99 (2) D-WARD (3) UNINA	Traffic traces (packet rates) and mean and std for the traces)	Generate anomaly profiles	Adaptive Threshold, Cumulative Sum, and Continuous Wavelet Transform
[114]	Network anomaly detection	DARPA99	15 features from TCP, UDP, ICMP data	ARX model using wavelet approximation coefficiets	DWT and AutoRegressive with eXogenous (ARX)
[115]	Network anomaly detection	(1) USC traces (2) Simulated virutal attacks on the University of Auckland traces	Duration, persistency, IP address	Coefficient-selective reconstruction in DWT	DWT and Threshold
[116]	Network anomaly detection	(1) DARPA99 (2) UNINA (3) D-WARD	Number of IP packets received	Euclidean distance between coefficients	Wavelet Packet Transform (WPT)
[109]	Network anomaly detection	Simulated dataset	Data packets	A scale-adaptive method	WPT
[117]	Classifying network applications	Simulated dataset	69 calculated features from TCP and UDP packets	Back-propagation (BP) neural network Particle and swarm optimization	Wavelet packet decomposition (WPD)
[118]	Detection of DoS attack	NLANR data	IP address, Packet size, Time stamp	Thresholding and wavelet variance computation	DWT

DARPA99: 1999 DARPA intrusion detection dataset
UNINA: University of Naples “Federico II” traffic trace dataset
D-WARD: UCLA Packet trace dataset

important information from the data. Therefore, DWT can detect anomalous patterns for identifying any unknown network traffic attack. Specifically, Wavelets are obtained from mother wavelet by dilations and shifting.

$$\varphi(t) = \frac{1}{\sqrt{a}}\varphi(\frac{t-b}{a}) \tag{1}$$

where a is a scaling parameter and b is a shifting parameter. Then, the wavelet is calculated with the signal;

$$\int_{-\infty}^{\infty} x(t)\varphi(t)\,dt, \text{where } x(t) \text{ is the original signal} \tag{2}$$

The ability to preserve both time and frequency resolution has led to widespread use of the DWT in many practical applications in other fields including Biology and Medicine. By applying DWT, hidden pattern that may be invisible to human eyes can be extracted. Therefore, DWT can detect hidden but important patterns for identifying network traffic data abnormality.

Although researchers [116, 109, 117, 113] utilized Wavelet Transform (WT) techniques in intrusion detection, they used WT only for detecting abnormal traffic by reconstructing the data or determining a threshold for detecting intrusions. This threshold would be a decision point to determine abnormalities in their studies. However, in this project, we used DWT to extract new features.

The selection of the specific mother wavelet is often considered as a difficult task since results can be various depending on what mother wavelet is applied. For this study, a broadly used Daubechies' wavelet family (Specifically, db2) is selected. A three-level decomposition is applied to the data with an overlapping sliding window to examine rapid changes within the data. A windows size (size of 100 data points) with a 75% overlap are tested. By applying DWT, three features (i.e. standard deviation of absolute values, root mean square, and energy) are calculated. The features are

$$\begin{aligned} \sigma_k &= \sqrt{[(\frac{1}{N}\Sigma_{i=1}^{N}(|d_i^k| - \mu)^2]}, \\ m_k &= \sqrt{((1/N \quad \Sigma_{i=1}^{N}(d_i^k)^2)}, \\ e_k &= \Sigma_{i=1}^{N}(|d_i^k|)^2 \end{aligned}$$

where $\mu = \frac{1}{N}\Sigma_{i=1}^{N} d_i^k$, N is the size of each coefficients, d_i is a wavelet coefficients, and k indicates decomposition level (i.e. $k = 3$). The extracted features are used as input features to generate a predictive model.

5.4.2 Feature selection using statistical analysis

Statistically significant features are determined. Among all the extracted features, less important and less irrelevant features need to be removed. Manually selecting features is not possible because it is difficult to select features with maintaining the correlation among them. In here, a statistical analysis is performed to determine the significance of each feature using Statistical Analytic Software (SAS). In particular, ANOVA analysis is performed and any features maintaining the statistical significance ($p < .05$) are selected. Among them, the most dominant features are used to generate a robust predictive model.

After extracting features from DWT, one-way analysis of variance (ANOVA) is used to find statistically significant features. The purpose of performing this ANOVA test is to identify significant differences among class means. ANOVA can be used to investigate whether the population means are the same. "F ratio", as an indicator of class separation, is computed from class variance over the within-class variance. The between class variance (sum of square variance among classes) is calculated as

$$\sigma_{bc} = \frac{\sum(x_i - \mu)\eta_i}{k-1}$$

Where η_i is the number of measurements in the ith class, $x_i$is the mean of the ith class, k is the number of class, and μ is the overall mean. The within class variance (within class sum of square variance) is calculated as

$$\sigma_{wc} = \frac{(\sum\sum(x_{ij} - \mu)^2) - (\sum(x_i - \mu)^2\eta_i)}{k-1}$$

Where x_{ij} is the ith measurement of the jth class. Then, the F ratio, F^{α} is calculated as the ratio between the two variances;

$$F^{\alpha} = \frac{\sigma_{bc}}{\sigma_{wc}} \geq F_{k-1,\sum_1^k(x_i-\mu)}(\alpha)$$

Where $F_{k-1,\sum_1^k(x_i-\mu)}$ is the upper (100α)th percentile of the F-distribution with $k-1$ and $\sum(x_i - \mu)$ degree of freedom.

5.4.3 Detection of exact attacks using machine learning

Once the features are extracted, the significance of each feature is tested. Only significant features are selected to generate a classifier (i.e. learning model) that can be used to detect exact attack categories using ML algorithms. Three ML algorithms such as SVM, Neural Network (NN), and Naïve Bayes are compared. Naïve Bayes and NN are commonly used to classify data consisting of two groups (e.g. normal/abnormal). The main idea of SVM, a statistical learning theory, is finding a hyperplane that can separate the input data precisely. That is, SVM finds the optimal hyperplane by minimizing the misclassification error. Naïve Bayes, a simplified Bayesian probability model based on bayes theorem,

calculated prior and conditional probabilities to generate a learning model. This learning model may cause an error because of the impacts of bias and variance, and data noise. NN is an information processing model that is inspired by the biological nervous systems. It is composed of a large number of highly interconnected neurons. It has limitations including falling into a local solution instead of global one and having a slow convergence. In general, SVM [121] is simple, fast in operation, and has good robustness than Bayes and Neural Network. Therefore, it is widely used in different domains such as bioinformatics [122], data mining, pattern recognition [123], and text categorization [124]. In this study, SVM is used to generate a classifier. Also, Logistic Regression (LR) is particularly useful when the class is dichotomous (e.g. normal/ abnormal) to measure the probability of classes. The logit function calculates the expected probability of a dichotomy as:

$$\pi_i = pr(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots)}} \tag{3}$$

where X_i is a variable with numeric value, $pi - i$ is the outcome (dichotomous; 0 / 1, e.g., normal/ abnormal), and the β_is are the regression coefficients that quantify the contributions of the numeric variables to the overall probability [125, 126, 127]. Unlike most regression analyses, LR does not need to assume data distributions on variables. Due to this benefit, it is commonly used when outcome is a nominal variable. A performance comparisons with the ML algorithms are conducted.

5.5 Designing a hybrid approach with computational analysis and visual analytics to detect network intrusions

5.5.1 Interactive visual analysis

A visual analytics approach is utilized to perform an interactive visual analysis of network traffic data. Visual analytics has been known as a new research area that focuses on performing analytical reasoning with interactive visual interfaces [128]. An extended version of our visual analytics tool (called iPCA [129]) is used to conduct an interactive factor analysis. iPCA is designed to represent the results of Principal Component Analysis (PCA) using multiple coordinated views and a rich set of user interactions to support interactive analysis of multivariate datasets. An extended version of iPCA contains five distinct views (see Figure 2). The network traffic data are projected onto two user-selected principal components in the Projection view (Figure 2A). In the Data view (Figure 2B), a parallel coordinates visualization is used to show all data in the original data dimensions. Horizontal lines represent features of the data and each line indicates each network traffic's data. Dimension sliders (Figure 2C) support controlling the amount of contribution of a dimension in PCA calculation. In the Eigenvector view (Figure 2D), a parallel coordinates visualization is also used to represent eigenvalues and eigenvectors. Each eigenvalue is treated as a dimension, and each vertical line represents each network traffic data's eigenvector. Lastly, Pearson-correlation coefficients and relationships (scatter plot) between each pair of variables are represented the Correlation view (Figure 2E).

Within iPCA, the user is allowed to select data in one coordinate space and immediately see

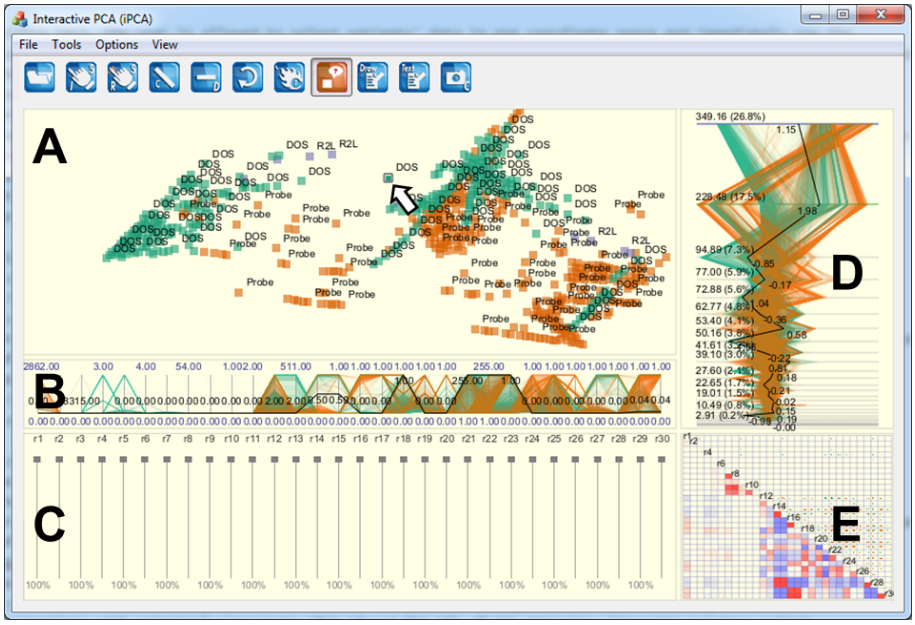


Figure 2: A visual representation of the NSL-KDD raw feature dataset with iPCA. It consists of five views - (A) Projection view, (B) Data view, (C) Dimension sliders, (D) Eigenvector view, and (E) Correlation view. The action of moving the mouse over the item (see the arrow) in the Projection view causes an event of highlighting it in other views. The arrow in the Projection view indicates that the user highlights a DoS attack to see its detail in other views (highlighted in black in (B) and (D)).

the corresponding data highlighted in the other coordinate space to help the user understand the relationship between the two. To enhance the capability of interactive visual analysis within each view, commonly known user interactions, i.e. highlighting, brushing, and filtering of data items or dimensions, are supported. In the Projection view (Figure 2A), additional user interactions including zooming and panning are supported to help the user navigate through the PCA projection space. It is important to note that whenever data modification is applied by removing data or adjusting dimension contributions, the re-computation of PCA is performed. A detailed explanation of the system can be found in [129]. Figure 2 shows the projection of the NSL-KDD raw feature dataset.

5.5.2 Reducing dimensionality of network intrusions

Multi-resolution analysis (MRA) utilizes signal processing techniques, so it is capable to discover hidden patterns from the network traffic data. Our approach utilizes DWT to extract features from the data in different resolution levels (γ). DWT is a broadly known time-frequency analysis method. It uses two basis functions such as wavelet function and scaling function [130]. The functions are applied to transform input data into a set of approximation coefficients and detail coefficients. Since a small localized wave in a time domain is analyzed, any sudden or rapid changes in the data can be identified. Thus, DWT is especially useful for analyzing non-stationary signal data such as internet traffic and network traffic data [131].

Sliding window analysis is a common approach when examining large network traffic data [132, 133]. The sliding window analysis uses two main parameters: window size and step size. The sliding window size (α) is used to extract feature vectors for analyzing anomalies in the network traffic data. If the size of the window is small, large feature vectors are generated. The step size (β) is considered as a tunable parameter that has a direct impact on identifying the anomalies. Since it indicates the distance between successive windows, if the step size increases, fewer windows are required to analyze the data. When applying DWT to network traffic data, an appropriate wavelet function should be chosen that is closely matched to variations in the input data. Also, choosing appropriate attributes to apply DWT to the network traffic data is critical [134, 135]. A common approach to determine the size of sliding window is referencing time information. However, there is no optimal approach of determining the attribute values for analyzing the network traffic data. Therefore, determining optimal values for sliding window (α), step (β), and level (γ) is important. In network anomaly detection with DWT, various levels of decomposition are often considered [136]. Depending on the decomposition level (γ), different levels of detail coefficients (detail level $1 \sim n$) and approximate coefficients can be measured. We performed an empirical study with MATLAB software to determine the optimal values for detecting network intrusion more precisely. In DWT, there are various wavelet families proposed by researchers such as Daubechies, Coiflets, Symlets, Discrete Meyer, Biorthogonal, and among others. To identify the best possible wavelet family for network intrusion detection, we evaluate all available wavelet families.

After identifying informative features with DWT, a statistical validation is performed to determine statistically significant features ($p < 0.05$). If the decomposition level is increased, the chance of having zero-variance features is also increased. Therefore, the statistical validation is useful for removing such unimportant features. Then, PCA is applied to determine principal components of the data. PCA computation is broadly used in feature extraction and exploratory data analysis in network intrusion detection [137, 138]. PCA performs eigenvalue decomposition to determine the variances and coefficients of the data by finding eigenvectors and eigenvalues. First, covariance is measured to determine how much the dimensions vary from the mean with respect to each other. Then, the eigenvectors and eigenvalues are calculated. The eigenvector with the highest eigenvalue is the most dominant principle component in the data, indicating the most significant relationship among the data dimensions. For this reason, PCA is often considered as a dimension reduction method for representing high-dimensional data into a lower dimensional space with the dominant principal components. To determine the principal components, we use Singular Value Decomposition (SVD) [139] because it is good for finding eigenvectors and eigenvalues in non-square matrice such as network traffic data. When representing data into 2D or 3D display space (i.e. coordinate system) with the principal components, confidence interval (θ) should be considered because it indicates the error between original and projected data. For instance, for mapping a high-dimensional data into a lower dimensional space, determination of an optimal low-dimensional space needs to be performed by considering the eigenvectors of the covariance matrix. However, if both the amount of data and the

size of attributes are large, identifying the best low-dimensional space is difficult due to the complexity of the data. Therefore, we use the first two principal components to display the network traffic data in a 2D display space with providing an option for the user to change them to others.

6 RESEARCH RESULTS

This section presents the generated rules to identify network abnormality, the performance of detecting exact attack categories, and the visual analysis to examine the relationship among the DWT features and its correlation analysis.

6.1 Rule generation to detect abnormal behaviors

As described in Section 5.3, total of 77,054 normal and 71,463 abnormal data are used. After converting the nominal input variables to binary scheme indicators, total of 90 variables including six binary variables are generated. A statistical analysis (i.e. ANOVA) is performed to determine statistically significant features. As a result, 22 features (e.g. ICMP, HTTP, SMTP, domain_u, SF, private, S2, S1, IRC, REJ, land_0, login_Yes, POP3, FTP, FTP_data, x11, Host_login_Yes,urp_i, Telnet, IMAP4, Guest_login_Yes, Gopher) are found to be statistically significant ($p < .05$). Then, the 22 significant features are used to generate decision trees. Ten trees are created and tested with distinctive test datasets. Table 6 represents the samples of extracted rules maintaining the testing accuracy of 85% or above.

Table 6: Samples of the extracted rules that are used to identify abnormal network traffic behaviors.

Rules	Testing Accuracy
If(SF='NO' & http='NO' & login_Yes='YES' & IRC='NO' & S1='NO' & smtp='NO' & X11='NO') then Abnormal	5521/ 5542=99.62%
If (SF='YES' & ICMP='YES' & urp_i='NO') then Abnormal	840/ 929=90.41%
If(SF='YES' & ICMP='NO' & private='NO' & pop_3='YES') then (Abnormal)	324/ 342=94.73%
If (SF='YES' & ICMP='NO' & private='NO' & ftp='NO' & pop_3='NO' & telnet='YES' & login_No='NO') then Abnormal	506/ 507=99.80%
if(SF='NO' & http='YES' & REJ='YES') then Normal	304/ 326=93.25%
If (SF='YES' & ICMP='NO' & private='NO' & pop_3='NO' & telnet='NO' & ftp='NO' & ftp_data='YES') then Normal	560/ 633=88.46%
If(SF='YES' & ICMP='NO' & private='NO' & pop_3='NO' & telnet='NO' & ftp='NO' & ftp_data='NO' & imap4='NO' & tcp='NO') then Normal	1271/ 1297=97.99%
If(SF='NO' & http='YES' & REJ='YES') then Normal	308 / 333=92.49%
If(SF='YES' & ICMP='NO' & private='NO' & pop_3='YES') then Abnormal	324 / 342=94.73%
If (SF='YES' & ICMP='NO' & Pop_3='NO' & telnet='NO' & ftp='NO' & ftp_data='NO' & imap4='NO' & tcp='YES' & login='NO') then Normal	4799 / 4913=97.67%
If (SF='YES' & ICMP='NO' & ftp='NO' & pop_3='NO' & telnet='NO' & ftp_data='NO' & gopher='NO' & login='NO') then Normal	5744 / 6085=94.4%

We found that "SF", one of the attribute values in "flag", is an important attribute to identify network abnormality. Also, the generated rules are complicated to present the "Abnormal" behavior. When considering the "SF" feature (indicating normal establishment and termination), if the "SF" feature is "NO", there is a higher chance that network activities are detected as abnormal behaviors.

However, it is important to verify the result by checking other features. Due to this reason, the size of the rule can be longer and complex than when the "SF" feature is "Yes".

6.2 Exact attack detection

To detect the exact attack category, thirty-two numerical variables in abnormal data (i.e. total of 71,344) are used. A total number of 54,275 data for the DoS attack, 14,077 for the Probe attack, and 2,992 for the R2L attack are used, respectively. Since two numerical variables (i.e. urgent and num_outbound_cmds) have all zero values, they are removed from the analysis. As explained in Section 5.4, DWT is applied to extract features. With the DWT, the total of 2,841 (2,167 for DoS, 559 for Probe, and 115 for R2L) datasets with 144 features are generated. A statistical test is applied to find a statistical significance of each feature. As a result, 77 out of 144 features were determined as statistically significant ($p < 0.05$) features.

6.2.1 Feature comparisons

A feature comparison between the raw and the DWT features is performed by measuring the average of the features. Since the raw and the DWT features have different scales, a normalization between 0 and 1 is applied. As shown in Figure 3, we found that the DWT features clearly separate the attack categories while the raw features maintain similar patterns. For the raw features, we noticed that the five features (i.e. r1, r4, r7, r8, and r14) are almost identical between the two attack categories (Probe and R2L). Although the DoS attack shows a distinctive pattern of the three attacks at the features (see the features of r5, r6, r10, r11, r12, and r13), the raw features may not be useful for differentiating the three exact attacks.

6.2.2 Visual comparison of the features

To project the raw and the DWT features, PCA computation is performed to identify principal components. PCA requires a high computational power to compute eigenvectors and eigenvalues. Thus an approximation method based on SVD called Online SVD [140] is used to perform the PCA computation and maintain real-time user interactions when interacting with large-scale datasets. Figure 4 represents PCA projections with two principal components on (A) the raw features and (B) the DWT features. From the projection of the raw feature (Figure 4a), it is difficult to identify a clear separation among the three attack categories. The DoS attacks appear mostly in three regions; the Probe attacks occupy two regions, and the R2L attacks are spread out all over the Projection space. It explains that identifying the difference among the three attacks is extremely difficult because they maintain similar patterns. However, there was a clear separation among the attacks in the projection of the DWT features (see Figure 4b). The DoS attack is forming two clusters that are completed separated from other attacks. Since there is a similarity between Probe and R2L even in the DWT features, additional analysis is conducted to determine common features appeared in both categories.

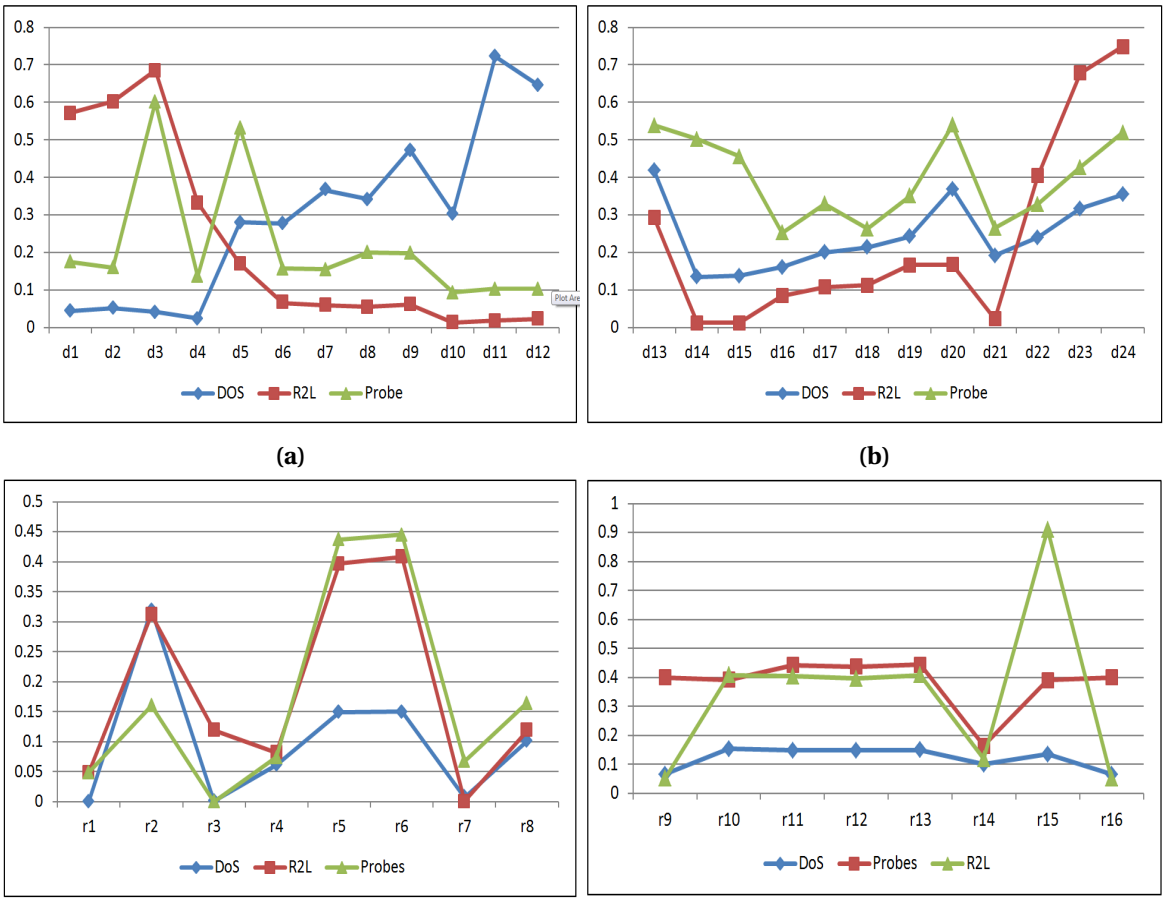


Figure 3: A comparison between the DWT features ((A) and (B)) and the raw features ((C) and (D)). x-axis indicates the DWT and raw features and y-axis presents the average value of each feature.

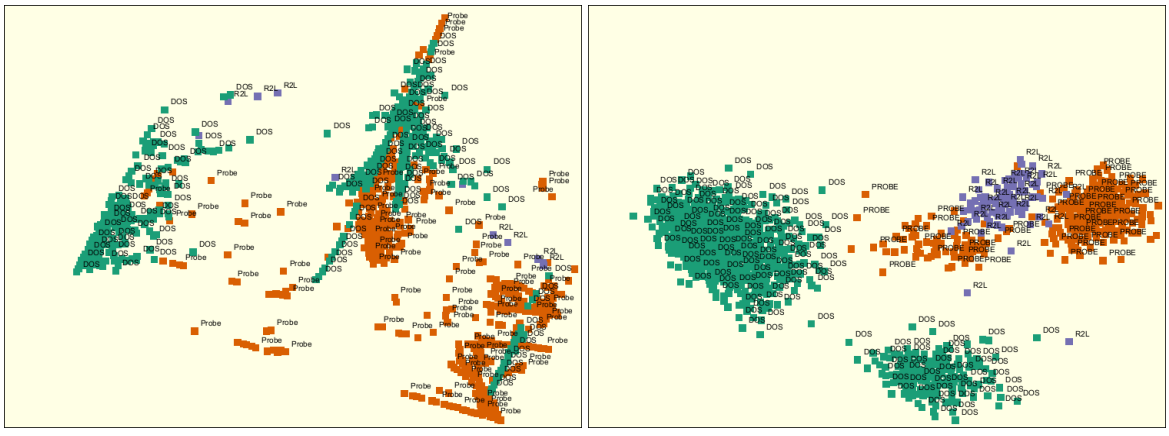


Figure 4: PCA projections of (A) the raw feature and (B) the DWT feature datasets. The data are mapped with different color attributes as DoS (green), Probe (orange), and R2L (purple)

6.2.3 Factor analysis

Dimension Contribution Analysis: With iPCA, interactive factor analysis is performed. Since iPCA supports the change of dimension contributions by moving the slider bar in each feature provides the ability to analyze the data non-linearly. When applying the dimension contribution, it is extremely important for the user to maintain an awareness of this change by the contribution since the projection of data will be modified. The user can easily become disoriented if the meaning of changes is unclear. For this dimension contribution change, there is a clear mathematical precedent to the use of dimension contributions. In Weighted Principal Component Analysis (WPCA), different variables can have different weights $s_1, s_2, ..., s_n$ [141]. It assumes that data are not always linearly increasing or decreasing, and there may be reasons to allow different observations to have different weights. Based on this assumption, WPCA is adopted by researchers when analyzing complex data to set different weights to each variable, to find missing data by giving zero weight to possible missing data, to create a nonlinear multivariate data analysis. As shown in Figure 5, when dimension contribution analysis is performed by changing the contribution of the five features (d37, d38, d68, d72, and d75) from 100% to 0%, a clearly separated pattern has appeared. As shown in Figure 5, a couple of possible outliers are visible. Figure 5 (A) indicates a R2L attack is appeared within a DoS cluster. Figure 5 (B) represents a DoS attack positioned in a R2L cluster. However, this result can be utilized to build a detection model to differentiate attacks since it maintains somewhat clear separation among the attack categories.

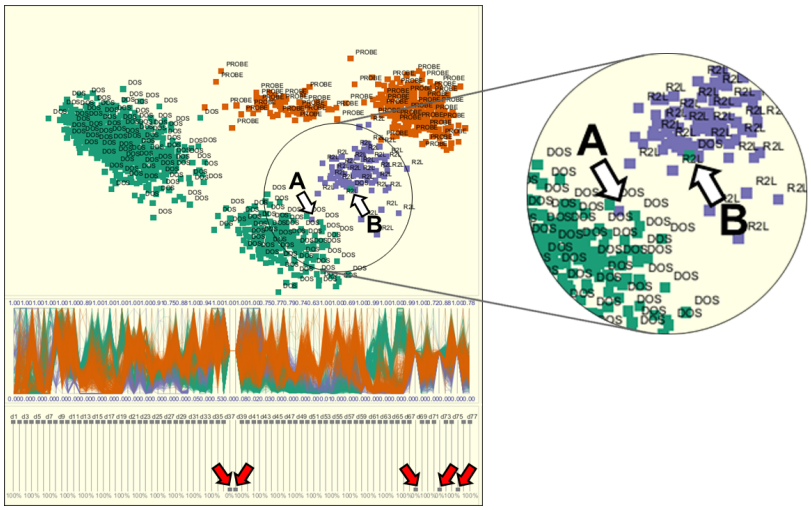


Figure 5: Dimension contribution is applied in the five DWT features (d37, d38, d68, d72, and d75) from 100% to 0% using the slider bars to make a clear separation between Probe and R2L (see the red arrows). 0% indicates that the selected variable is not used to going to contribute to the final PCA.

To investigate the relationship among the features, Pearson-correlation analysis between each pair of features is conducted. Figure 6 represents the correlations of the (A) raw and (B) DWT feature datasets. In Figure 6, the diagonal displays the name of dimension as a text string. The lower triangulation shows the coefficient value between two dimensions with a color indicating positive (red),

neutral (white), and negative (blue) correlations. The upper triangulation contains cells of scatter plots where all data items are projected onto the two intersecting dimensions. As we discussed above, there was no clear separation among the attacks using the raw features (see Figure 4a). This might be because a half of the features maintain neutral correlations (Figure 6a). However, positive and negative correlations are easily discovered in the DWT features (Figure 6b). When looking at the scatterplots having highly positive correlation coefficients ($\gamma = 0.99$) in Figure 6c and 6d, we identified that they maintain different distributions. Although the scatterplot in Figure 6c shows vertically or horizontally increasing patterns (i.e. skew correlation), the scatterplot in Figure 6d presents a directly proportional pattern by showing a linear relationship between the two features. Also, the scatterplot (Figure 6d) displays that the attack categories are appeared by forming different patterns as the R2L attacks are mostly appeared in the lower bottom corner, the DoS attacks are forming two visible clusters, and the Probe attacks are spread out in the middle and lower regions.

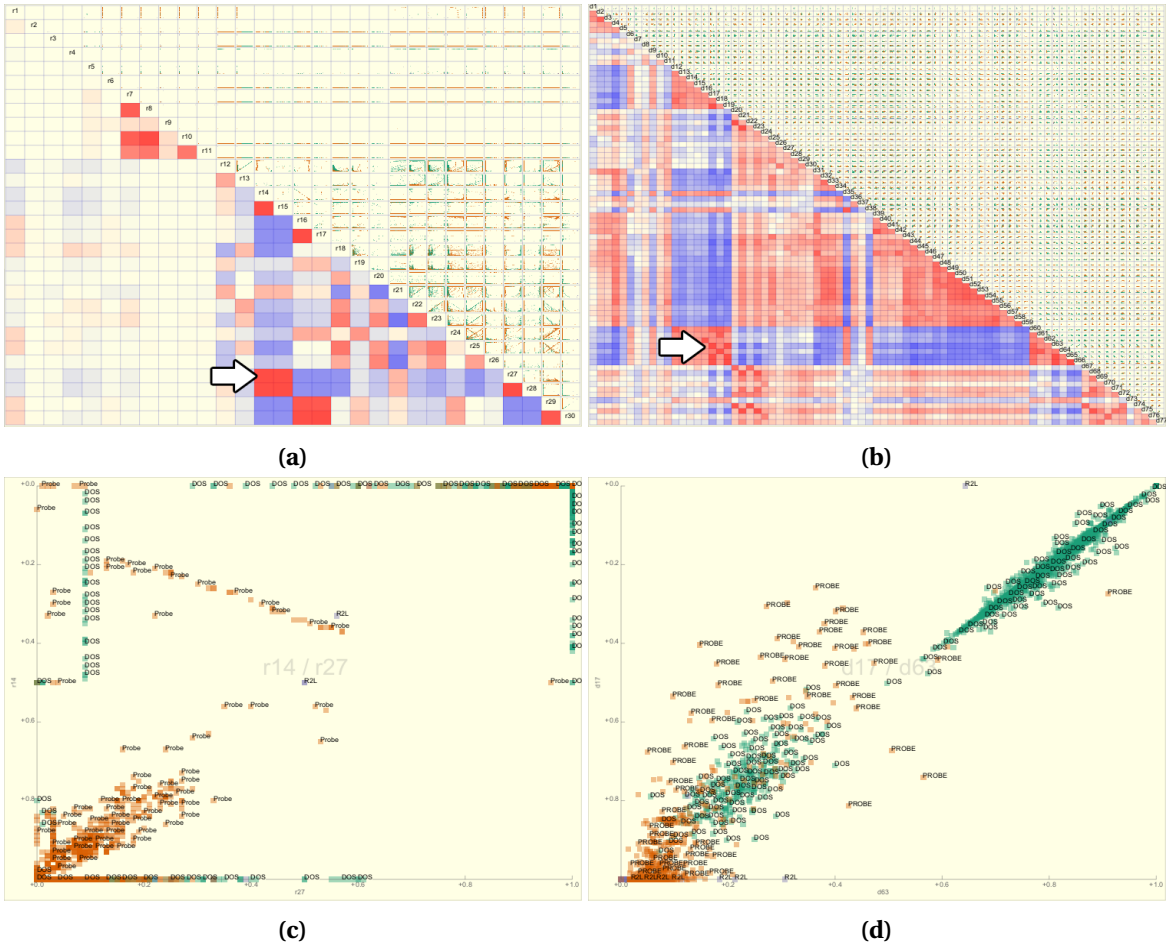
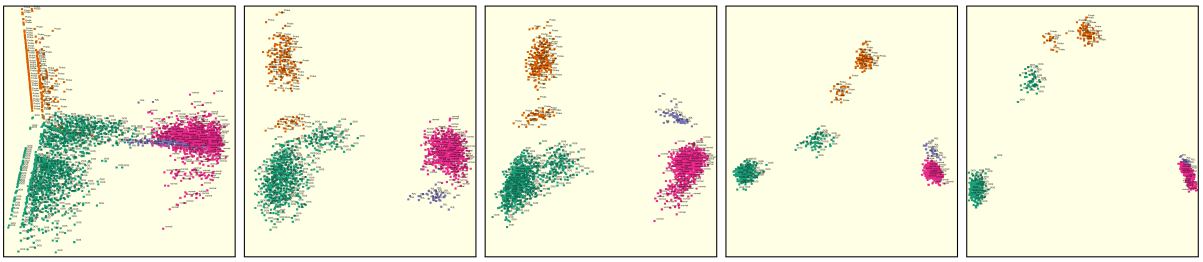


Figure 6: Correlation views of the (A) raw feature and (B) DWT feature datasets. Each color indicates positive (red), neutral (white), and negative (blue) correlations. The arrows in (A and B) indicate the scatterplots having positive correlation coefficients ($\gamma = 0.99$). Their scatterplots are presented in (C and D).

6.2.4 Sliding window with different wavelet comparisons



(a) $\alpha:50, \beta:25, \gamma:3$ **(b)** $\alpha:125, \beta:75, \gamma:3$ **(c)** $\alpha:150, \beta:50, \gamma:3$ **(d)** $\alpha:200, \beta:100, \gamma:3$ **(e)** $\alpha:300, \beta:150, \gamma:3$

Figure 7: An example empirical study result of identifying optimal values for sliding window size (α), step (β), and wavelet level (γ) with the wavelet function (i.e. Daubechies 3) to detect network intrusions.

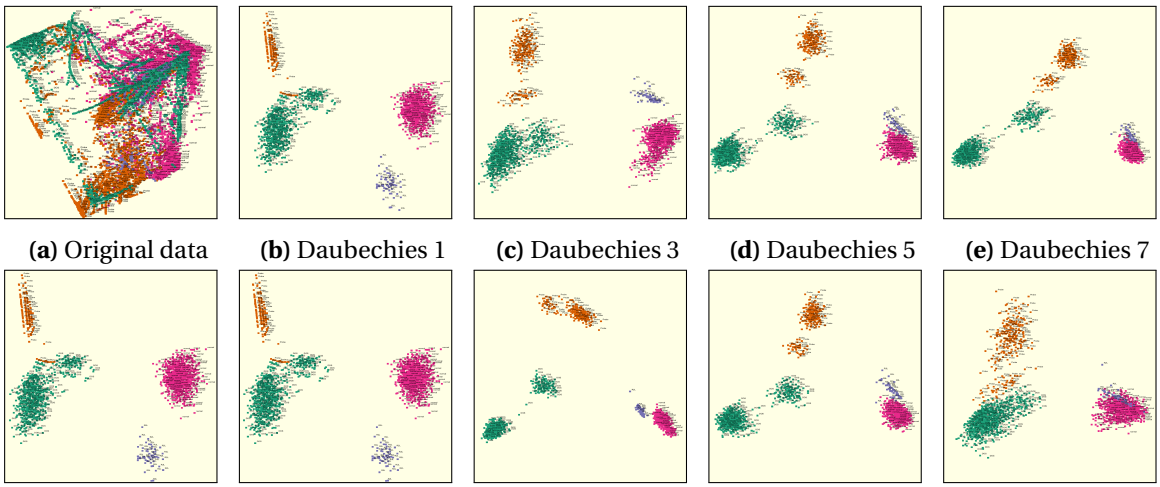
An empirical study was performed to determine the optimal values for DWT (Figure 7). For the sliding window (α), different window sizes ranging from 25 to 250 were tested. Depending on the size of sliding window, different step size (β) was applied. For instance, if the sliding window size is set to 250, the step sizes from 25 to 225 are tested. Since DWT is influenced by the decomposition level, determining appropriate decomposition level (γ) is also important [142]. Based on the analysis, we identified that the optimal values for detecting network intrusions are $\alpha = 150$, $\beta = 50$, and $\gamma = 3$. We also found that if the sliding window size is decreased ($\alpha < 150$), the chance of having false positives increases. However, if the window size is increased ($\alpha >= 150$), a clear separation has emerged among the attacks. Interestingly, we found that if the size of the window is increased continuously, it tends to remove unique characteristics of the attacks. For instance, as shown in Figure 7d and 7e, Normal and R2L attacks are appeared in the same cluster when the size of the sliding window is increased ($\alpha >= 200$). A further study needs to be performed to understand the effectiveness of the sliding window size.

When applying DWT with different wavelet functions and levels, different amounts of data records and attributes are generated. For instance, when applying Daubechies 3 (i.e. db3) with $\alpha = 150$, $\beta = 50$, and $\gamma = 3$, total of 2,958 data records with 703 attributes are generated. Out of 703 attributes, 418 (59.5%) are selected as statistically significant attributes ($p < 0.01$). However, when using Discrete Meyer with $\alpha = 50$, $\beta = 20$, and $\gamma = 3$, total of 7,417 data records with 3,478 attributes are generated. And, 2,992 attributes (86%) are determined as significant features ($p < 0.01$). The significant features are considered as dominant features that can be used to detect intrusions. If the size of the window (α) is small, large feature vectors are generated. The sliding window size can be adjusted to speed-up the computation process. However, it may compromise the overall accuracy of detecting intrusions if improperly set. It is also important to note that if the step size (β) is small, more windows are required to analyze the data.

With the original network traffic data, it is difficult to determine normal vs. attack activities. As we discussed, MRA determines significant features from the data. However, it is still not clear what DWT wavelet families produce more significant features to detect intrusions. Therefore, we tested

most wavelet families under the same experimental condition (i.e. $\alpha = 150$, $\beta = 50$, and $\gamma = 3$). Figure 8a shows a PCA projection of the original data. As can be seen, different network activities appear in all over the place which makes us difficult to separate them clearly. On the other hand, when applying different wavelet families, a somewhat clear separation between normal and attack activities is appeared by forming unique clusters (see Figure 8b ~ 8j). Biorthogonal (Figure 8f and 8g) shows that some Probe attacks appear in the DoS attack cluster. Discrete Meyer (Figure 8h) and Symlets (Figure 8i) are suitable for separating DoS and Probe attacks from R2L attack. But, R2L attack is appeared near to the normal network activity cluster (see the regions in left-bottom in Figure 8h and 8i). Another interesting observation is that the results of Coiflets 1 and Daubechies 3 are similar to each other. The reason might be because Coiflets is constructed from Daubechies with a high number of vanishing moments in the scaling and the wavelet functions [143].

Huang et al. [135] found that Coiflet and Mexican Hat wavelets are good for detecting anomalies when using a five-minute, sixty-sample window. However, we identified that Daubechies 3 is the best-suited wavelet for detecting network intrusions since it creates well-separated clusters (see Figure 8c). Huang's experiment was well formatted, but they used only the first and second coefficients based on the assumption that these two coefficients have sufficient information. As they commented, any larger coefficients might contain too sparse information. However, from our empirical study, we found that the coefficient ($\gamma = 3$) is better for extracting significant information to detect intrusions.



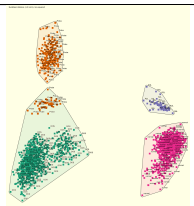
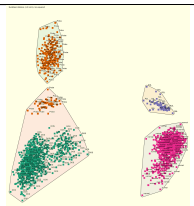
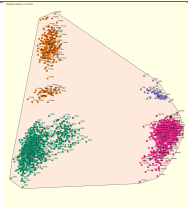
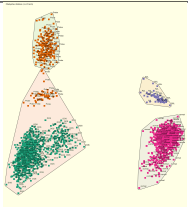
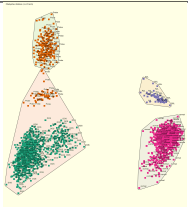
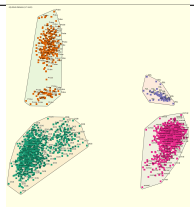
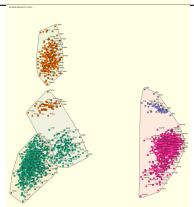
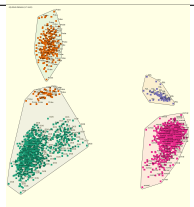
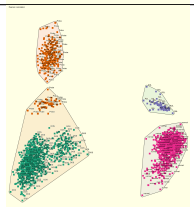
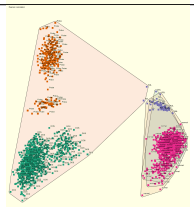
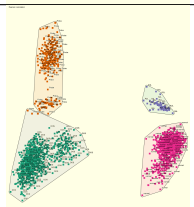
(a) Original data **(b)** Daubechies 1 **(c)** Daubechies 3 **(d)** Daubechies 5 **(e)** Daubechies 7

(f) Biorthogonal 1.1 **(g)** Biorthogonal 2.2 **(h)** Discrete Meyer **(i)** Symlets 5 **(j)** Coiflets 3

Figure 8: PCA projections of (a) original data and (b) DWT features with various wavelet families.

Table 7 shows the results when the hierarchical clustering method is applied to three different datasets: wavelet features (3,478 attributes), two principal components (2 attributes), and three principal components (3 attributes). The results are generated with the DWT features ($\alpha = 150$, $\beta = 50$, and $\gamma = 3$). Solid connected lines indicate the clustering results. Since the wavelet features are the statistically validated DWT features, good clustering results are generated. However, we found that a large number of attributes often require a significantly large computational time which is not

Table 7: Results of the detected k clusters with different distance metric (Euclidean distance (L^2), Chebyshev distance (L^∞), City-block distance (L^1) and Pearson correlation coefficient (R^2)). The cluster are represented as solid connected lines

Metric	Wavelet Features	Two Principal Components	Three Principal Components
L^2			
L^∞			
L^1			
R^2			

appropriate for the real-time intrusion detection. To support the real-time intrusion detection, PCA is considered because it has a benefit of reducing the total number of attributes (i.e. dimensions). As mentioned above, the confidence interval (θ) is often considered to determine major PCA components. However, minor PCA components (i.e. after removing the components that have higher eigenvalues) are also essential for revealing anomalies in network traffic data [144]. Because of this reason, allowing the user to select different PCA components is necessary. Our visualization tool has the feature of selecting the best possible PCA components to detect intrusions.

From the hierarchical clustering results, clustering accuracy is measured as 0.86 ± 0.16 (for using all wavelet features), 0.64 ± 0.04 (for using two principal components), and 0.83 ± 0.18 (for using three principal components). When comparing the results of the two vs. three principal components, the three principal components produced a better clustering result. Moreover, the three principal components and the wavelet features showed similar clustering accuracy. We also found that the three principal components outperform the other two when Chebyshev distance (L^∞) is applied.

Interestingly, Chebyshev distance (L^{∞}) was not good for creating clusters with the wavelet features. Moreover, Pearson correlation coefficient (R^2) did not work well for creating clusters with the two and three principal components.

6.2.5 Classification comparisons

Classification comparison using the Internet Traffic dataset: Table 8 indicates that the predictive model (with the DWT features) showed a better performance than with the raw features. In particular, the model specificity using the raw features was much higher than the DWT features in all four sliding windows. From the reliability measure (i.e. AUC), we found that the predictive model with DWT features provides a better ability of detecting abnormal internet traffic behaviors. We also found that the reliability of the predictive model with the DWT features is almost linearly increasing when the window size is incremented.

Table 8: A standard error mean comparison of the predictive model using the raw and the DWT features.

	the DWT features				the raw features
window size (i.e. data points)	25	50	100	150	-
Accuracy	1.107	0.611	0.315	0.150	1.529
Sensitivity	0.712	0.635	0.181	0.073	0.408
Specificity	1.810	0.525	0.275	0.087	3.086
AUC	0.377	0.409	0.044	0.040	0.692

Since it is important to perform a comparison with other approaches to determine the effectiveness of our proposed model, we conducted a study with broadly known techniques, such as Neural Network (NN) and Support Vector Machine (SVM). Specifically, we measured accuracy, sensitivity, and AUC (see Table 9). From the performance comparison, we identified that the accuracy, sensitivity, and AUC of our method were higher compared to other approaches (NN and SVM). This explains that our predictive model (via LR) is good for identifying abnormal behaviors more accurately. In addition, we also found that utilization of the DWT features is important for increasing the performance of detecting anomalous network traffic or behaviors.

Table 9: A performance comparison between our proposed method (using LR) and other broadly known approaches (SVM and NN). Each value indicates mean ± standard error mean (SEM).

		Accuracy	Sensitivity	AUC
With Raw Features	Logistic Regression (LR)	84.1 ± 1.52	95.9 ± 0.40	80.3 ± 0.69
	Neural Network (NN)	75.4 ± 1.73	81.3 ± 2.04	81.6 ± 1.94
	Support Vector Machine (SVM)	76.4 ± 2.81	75.4 ± 3.23	75.6 ± 2.82
With DWT Features	Logistic Regression (LR)	97.6 ± 0.61	97.8 ± 0.63	98.6 ± 0.40
	Neural Network (NN)	96.7 ± 0.51	96.7 ± 0.51	96.7 ± 0.55
	Support Vector Machine (SVM)	83.9 ± 5.21	85.2 ± 5.69	89.4 ± 4.75

Classification comparison using the NSL-KDD dataset: A classification is performed to deter-

mine exact attack categories with a ten-fold cross-validation (CV). The performance of three ML techniques (i.e. SVM, Naïve Byes, and NN) is compared and presented in Table 10. The average accuracy to detect exact attack categories with SVM, Naïve Bayes, and NN were 95.5471%, 89.024%, and 96.67%, respectively. We found that NN shows a slightly higher accuracy than SVM. However, when measuring the standard error of the mean (SEM), there was a variation difference as SVM (0.285), Naïve Bayes(2.02), and NN (0.683). When generating a learning model with SVM and NN, it took 0.157 seconds and 13.04 seconds, accordingly.

Table 10: Classification performance comparisons

	Three attack classification		
	SVM	Naïve Bayes	NN
Test 1	95.77%	91.47%	94.77%
Test 2	96.83%	91.23%	95.93%
Test 3	96.83%	95.5%	100%
Test 4	95.77%	89.77%	96.83%
Test 5	96.47%	90.47%	95.77%
Test 6	96.12%	78.2%	96.1%
Test 7	95.77%	89.1%	94.57%
Test 8	95.77%	93%	94.2%
Test 9	97.88%	76.8%	100%
Test 10	98.22%	94.7%	98.59%

6.3 Survey results on utilization of ML algorithms in cloud computing

6.3.1 Background

The introduction of cloud computing provided a dramatic change in data management and processing across many different fields of computing - not only does it shift infrastructure and computation to the network, it also dramatically reduces costs associated with the management of hardware and software resources. Furthermore, it has resulted in the development of new programming models such as MapReduce (e.g. Hadoop), BigTable, and hybrid systems like Hive for analyzing large, complex, and disjointed data sets [145]. Using these models, numerous studies have been performed to conduct computation analysis in cloud computing [61, 146, 147].

As we have discussed above, many machine learning algorithms can be paired with cloud computing technologies to improve intrusion detection analysis. When dealing with massive amounts of data (on the order of terabytes or petabytes, for example), intrusion detection becomes extremely difficult, and single computers cannot handle the sheer size of data. Thus, we turn to the cloud. Recent cloud-based intrusion detection techniques have predominantly employed the MapReduce model, an abstract programming model that processes large datasets on clusters of computers. MapReduce is composed of two somewhat obvious steps - map and reduce. In the MapReduce model, a large dataset is split and each split sent to a node, also known as a mapper, where each split is independently processed. Mapper results are then shuffled, sorted, and passed to reducers that digest and prepare

the final results [59]. A possible implementation of MapReduce for a cloud-based intrusion detection technique is inherently simple. The map step examines each split for traffic anomalies, and the reduce step combines them, packages them, and presents the overall report. Many researchers' intrusion detection techniques follow this general approach [59, 148, 61]. While particulars change, MapReduce remains a foundational tool throughout the most recent research on cloud-based intrusion detection analysis. Throughout the research, these implementations greatly reduce computing times for large traffic datasets.

Nonetheless, moving these intrusion detection techniques to the cloud brought with its own unique set of challenges, most visible of which was the unsuitability of certain algorithms to cloud server architecture. Important and popular algorithm (such as the aforementioned k-means algorithm) cannot be directly implemented in the MapReduce framework due to iterative computations that reference all input data. As MapReduce splits input data to be processed among numerous computers, the algorithm cannot access inputs on different computers in the cloud architecture. However, researchers [72] adapted the k-means algorithm for use with MapReduce, by constructing and sharing a global array of centers that allow all distances to be calculated. As a popular cloud-based machine learning and data mining tool built on MapReduce, Mahout [149] and MLlib (aka SparkML) [150] support various MLAs including k-means functionality. However, writing new or customizing existing algorithms is too costly because all algorithms need to be implemented (or modified) by following fixed distributed runtime plans and underlying data-parallel framework. To address this limitation, researchers proposed several approaches of fast implementing approaches as SystemML[151], NIMBLE[152], MLbase [153], Distributed GraphLab [154], and Tupleware [155]. These approaches are classified as "Declarative ML" [156]. Declarative ML simplifies the development of MLAs by separating algorithm semantics from underlying framework and execution plans to make them run in a cloud computing environment more efficiently. Since Declarative ML is a rather new approach, it has not been broadly used to intrusion detection study yet.

Table 11 shows login attempts to the CSITCLOUD system at the University of the District of Columbia. Given a large login attempt dataset, attempts per port numbers can be traced within the MapReduce paradigm. First, a mapper grabs all port number data and places them into a key-value pair, and then a reducer condenses the data into more discrete, manageable sets (i.e. Port 22: 2; Port 80: 5, etc.) In general, the reducer performs its job in three phases: shuffle, sort, and reduce. The reduced dataset can then be processed by MLAs to classify port activities, as was shown in [76].

6.3.2 Network flow and feature selection

While the MapReduce paradigm can be effective for larger datasets, some algorithms still have trouble swiftly dealing with large volumes of continuously generated traffic data. To combat this, some researchers have turned to network flow applications (such as NetFlow, sFlow, OpenFlow and IPFIX) to cut down on data overheads with filtering, sampling, and flow aggregation techniques [157].

Lee et al. [158], for example, used Cisco NetFlow to monitor internet traffic, filtering unnecessary

Table 11: An example of login attempts to the cloud computing system (called CSITCLOUD).

May 31 08:17:01 csitcloud CRON[22]: pam_unix(cron:session): session opened for user root by (uid=0)
May 31 08:17:01 csitcloud CRON[22]: pam_unix(cron:session): session closed for user root
May 31 09:17:01 csitcloud CRON[80]: pam_unix(cron:session): session opened for user root by (uid=0)
May 31 09:17:01 csitcloud CRON[80]: pam_unix(cron:session): session closed for user root
May 31 09:34:07 csitcloud sshd[25]: reverse mapping checking getaddrinfo for 197.51.174.61.dial.wz.zj.163data.com [61.174.51.197] failed - POSSIBLE BREAK-IN ATTEMPT!
May 31 09:34:07 csitcloud sshd[25]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=61.174.51.197 user=root
May 31 09:34:10 csitcloud sshd[25]: Failed password for root from 61.174.51.197 port 7370 ssh2
May 31 09:34:21 csitcloud sshd[25]: Disconnecting: Too many authentication failures for root [preauth]
May 31 09:34:21 csitcloud sshd[25]: PAM 5 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=61.174.51.197 user=root
May 31 09:34:21 csitcloud sshd[25]: PAM service(sshd) ignoring max retries; 6
Aug 31 09:34:23 csitcloud sshd[80]: reverse mapping checking getaddrinfo for 197.51.174.61.dial.wz.zj.163data.com [61.174.51.197] failed - POSSIBLE BREAK-IN ATTEMPT!
Aug 31 09:34:24 csitcloud sshd[80]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=61.174.51.197 user=root
Aug 31 09:34:26 csitcloud sshd[80]: Failed password for root from 61.174.51.197 port 15348 ssh2

data out of flow records using MapReduce to improve computation times by 72%. Li et al. [76] combined sFlow, MapReduce, and different MLAs (SVM and Decision Tree) to successfully classify host roles-a critical component of intrusion detection.

Nonetheless, while these network flow applications offer random and deterministic packet sampling, as well as another filtering, feature selection, and aggregation tools, they are often not sufficient for the complex world of cloud-based intrusion detection. As such, in recent years, researchers have increasingly turned to more in-depth, algorithm-based feature selection techniques to improve the results of their cloud-based intrusion detection techniques.

In particular, while network flow applications do provide some feature selection techniques, the problem of which features to choose remains a critical one for intrusion detection techniques on the whole. Not every feature of the data will be relevant, and some features can introduce noise or redundancy. Thus, selecting the optimal subset of features has become of great interest to researchers in recent years, and MLAs have provided an effective way to zero in on the best feature choices.

Stein et al. [159] pioneered this work before the cloud, improving Decision Tree classification performance by introducing genetic algorithm-based feature selection that eliminates distracting or unnecessary features. In the cloud, Muthurajkumar et al. [148] employed a Rough Set-based feature selection algorithm that generates feature subsets designed to find the best balance between detection rates and false alarms in a cloud-based intrusion detection technique. Chen et al. [160] used a MapReduce-based implementation of the OneR classifying algorithm, alongside vertical compression to improve detection up to 184 times with only tolerable losses in performance in their SVM-based cloud intrusion detection technique.

Although these feature selection algorithms have improved their respective intrusion detection techniques, they are not without their drawbacks. In some cases, while feature selection may improve the speeds of the detection algorithms, the overall time run-time increases, and training data cannot

be incrementally handled [159]. Thus, if feature selection algorithms have to periodically re-assess optimal feature subsets, they may provide even more overhead, or miss newer, more sophisticated attacks. Chen et al. [160] argued that these drawbacks can be improved by refining MapReduce performance and implementing newer algorithms with incremental clustering (or classifying) abilities.

6.3.3 Implementation examples

With the MapReduce paradigm, intrusion attempts per port number can be traced by analyzing a large login attempt dataset. First, a mapper grabs all port number data and places them into a key-value pair, and then a reducer condenses the data into more discrete, manageable sets (i.e. Port 22: 25 entries; Port 80: 50 entries, ...) In general, the reducer performs its job in three phases: shuffle, sort, and reduce. The reduced dataset can then be processed by ML algorithms to classify port activities, as was shown in [76]. Theoretically, when considering the analysis of the login attempts, the input data can be formatted to become a set of key-value pairs as (k_i, v_i) and the map function is applied to produce a list of intermediate key-value pairs as $map : (k_i, v_i) \rightarrow list(k_j, v_j)$. Since the intermediate list of key-value pairs indicates the outputs produced by mappers, they need to be merged by reducers as $reduce : (k_2, list(v_2)) \rightarrow list(k_3, v_3)$. This mechanism is also efficient for detecting anomalous network activities by analyzing IP flow records [60, 161].

When utilizing cloud computing architecture for intrusion detection, most cloud-based intrusion detection techniques are designed consisting of multiple components as data parser, data processing, data mapper and reducer. The data parser extracts essential information from the input data by eliminating unnecessary data. It mainly focuses on getting rid of useless (or redundant) features (i.e. variables) as well as identifying unknown but significant features from "Big Dimensionality" data [162]. The parsed information is then processed to determine important features, which are formatted as metadata file and distributed to HDFS nodes. Then, cloud job dispatcher launches the data mapper to assign jobs to each computing node. After completion of the mapping process, the data reducer is performed to reduce redundancy information by merging them. The MapReduce model can be adapted to run each component. For instance, the model is often used to extract features since it requires a longer processing time [163]. Most supervised MLAs requires separate training and testing datasets. With the training dataset, a learning model is generated. Then, the learning model is applied to validate and test with the testing dataset to show the effectiveness of the generated model. To run the supervised learning algorithms in the cloud, researchers proposed an idea of conducting the model generation as a sequential, but parallel processing on testing and training the data for detecting intrusions [164, 165].

As we discussed above, ML algorithms are generally not the sole component of a functioning intrusion detection, often working in tandem with other algorithms. Below we outline some of the most salient examples of MLA implementation in cloud-based intrusion detection techniques in recent research, and how they fit in to their respective systems.

- Muthurajkumar et al. [148] introduced an intrusion detection model that used a combination of

fuzzy SVM and feature selection algorithms to produce high detection rates and minimal false positives.

- Vieira et al. [166] proposed a Grid and Cloud Computing Intrusion Detection System (GCCIDS) that combats attacks by using both signature-based and anomaly-based techniques to detect intrusions. In order to train the system, the authors employed neural network classification algorithms, and the resulting system boasted low processing overhead and satisfactory performance for real-time implementation.

- Singh et al. [167] implemented rForest for peer-to-peer botnet detection that proved adept at classifying malicious traffic on a cluster, with low false positive rates and considerable precision and recall.

7 DISCUSSION & CONCLUSION

Understanding network traffic is important for securing our computing infrastructures. However, it is difficult to differentiate abnormal network behaviors from normal network traffic. This research presents a multi-level network abnormality detection method by utilizing reliable rules to detect abnormal behavior, generating a predictive model to detect the exact attacks (i.e. DoS, R2L, and Probe) using the DWT-based features, and displaying them into a visual analytics tool to provide further detailed understanding and analysis for users. Although DWT was often used by researchers to detect network abnormal behaviors, it was simply used to determine a threshold or to reconstruct data by removing noise. Unlike other studies, this project emphasizes the importance of using DWT to extract significant features for detecting network abnormal behaviors.

We introduced a rule-based approach to detect network abnormality. The most critical advantage of adapting rule-based method is that transparent rules can provide reasons (i.e. detailed explanations) behind the predictions. Thus, we generated rules with CART with only four variables (i.e. duration, protocol type, service, and flag) to detect four attack categories. The rules were statistically significant to detect intrusions. While the generated rules clearly differentiated normal and abnormal behaviors, there was a limitation of providing a detailed information about the detected abnormal behaviors since each variable includes numerous attribute values. For instance, the rule ($protocol \neq HTTP$) does not provide useful information because there are about 70 attribute values indicating different network protocols. To avoid this ambiguity, the nominal variables are converted dummy variables to generate more accurate rules. So, the result can provide appropriate meaning about the detected network abnormal behaviors. Each generated rule (see Table 6) presents a simple and specific information. Based on the performance measure of each rule, only highly accurate rules were used for intrusion detection analysis. However, it is important to note that even the rules with less accuracy may provide a valuable information for detecting intrusions. For instance, the rule - if (SF='YES' & ICMP='NO' &

Table 12: Network intrusion detection techniques that have been developed utilizing cloud-computing technology.

Work	Goal	Dataset(s)	Major Approaches	Cloud Environment	ML Algorithm(s)	Advantages	Challenges
Lee et al. [158]	Monitor internet traffic flow	Simulated NetFlow packets	(1) Packet sampling, (2) Flow aggregation, and (3) MapReduce programming model	Apache Hadoop	None	Flow computation time improved by 72% over legacy tools	Batch-processing jobs and text input file formats difficult to handle; flow analysis tools are not adequately developed for the MapReduce interface
Singh et al. [167]	P2P botnet detection	Simulated and CAIDA sample datasets	(1) Information Gain measurement and (2) Clustering (Random Forest) in Mahout	Apache Hadoop	Random Forest	Process high bandwidth in quasi-real-time, effectively classifies malicious traffic on a cluster	High packet drop rates, detection times still a little too high, cannot respond to newer, more sophisticated threats
Bhat et al. [168]	Anomaly intrusion detection	NSL-KDD 99	(1) Naïve Bayes (NB) Tree and (2) A hybrid approach of NB tree and Random Forest	Amazon EC2	NB Tree and Random Forest	Good performance, high accuracy, low false positive rate for NB Tree / Random Forest hybrid implementation	High false positive rate for non-hybrid implementations
Chen et al. [147]	Phishing attack detection	Simulated dataset	Apache Hadoop	Eucalyptus, Apache Hadoop, and Amazon EC2	Collaborative algorithm based on Distributed Hash Tables (DHT)	Practical scheme, can be generalized to other attacks	Not tested with various datesets
Chen et al. [160]	Intrusion Detection	KDD 99, CMDC 2012	(1) Feature Reduction, (2) Vertical compression, and (3) Intrusion detection	Apache Hadoop	OneR algorithm, Affinity propagation, KNN, and SVM	Faster than traditional models	No incremental clustering ability - feature reduction and training steps can provide significant overhead
Marnerides et al. [169]	Malware detection	Simulated dataset	(1) Energy estimation, (2) Feature Selection, and (3) Covariance Analysis	Unknown	Choi-Williams Distribution	Effective for identifying Kelihos injection	Not tested with various datesets
Muthurajkumar et al. [148]	Intrusion detection	Simulated dataset	(1) Feature selection and (2) Fuzzy SVM	Unknown	Rough Set based Feature Selection Algorithm (RSFSA), Fuzzy SVM	Reduces number of decision attributes and the size of log data, faster than traditional models	Not tested with various datesets
Vieira et al. [166]	Intrusion detection technique	Simulated dataset	Utilization of Grid and Cloud Computing	Unknown	Feed-Forward Neural Network	Successfully explores communication events to mark intrusion	Large sample period of data is required and training cannot adapt new threats
Wang et al. [170]	Network traffic passive measurement	CAIDA dataset (anonymized traffic data collected from equinix-chicago and equinix-sanjose)	IP Trace Analysis System (IPTAS)	Unknown	None	Useful prototype of passive traffic analysis tool	Not provide a fine-grained traffic analysis

private='YES') then Abnormal - has 72.16% of accuracy. Although the accuracy does not represent a high performance, we found that the rule is fitted to the majority of the data (306 / 424).

Also, we proposed a predictive model to identify exact attack types of abnormal network traffic. Instead of using the raw feature for the predictive model, Higuchi fractal dimension and statistical measures are applied to extract significant features in identifying four attack categories (i.e. DoS, Probe, U2R, R2L). Prior to applying them directly to design the predictive model, all extracted features were analyzed to determine their statistical significance. Also, we found that the performance accuracy of detecting network anomaly was high when using the extracted features. In this study, we considered performing a comparison between the SVN-based predictive model and a broadly used NN-based predictive model. From this comparison, we identified that SVN-based predictive model shows a better performance than the NN-based model. Although the overall accuracy of the proposed model was 77.1%, true positive of DoS and Probe attacks showed over 90% of accuracy.

Among the extracted DWT features, 53.47% features are shown to be statistically significant ($p < 0.05$). Even though R2L attacks have less amount of data compared to other attacks, we identified that the true positive for the R2L with the raw feature is 59.8 % and 75% for the DWT features. One of the major concerns in many previous studies for detecting intrusions is how to reduce high false positive (FP) results. In our study, the FP rate for the raw and the DWT features were 7.9% and 2.3%, respectively. The DWT features can provide a better performance if we have a larger amount of R2L data. It is also important to note that, unlike other previous methods utilizing wavelet transform techniques, our approach includes a method of performing a mathematical calculation and a statistical validation to extract hidden underlying patterns from the input data.

In this project, we also utilized a visual analytics tool to interpret the results, discover new knowledge, and find reasons efficiently. As shown in Figure 4, there was no clear separation of the raw features among DOS, Probe, and R2L. However, when using the DWT features, we identified a clear separation among the attack categories. Most importantly, the "R2L" attack was not identifiable with the raw features. When analyzing the DWT features further, we identified that there was a similarity between Probe and R2L. The dimension contribution analysis was performed with iPCA to identify specific features that make them difficult to separate. The dimension analysis with iPCA is quite challenging because the user needs to maintain an awareness of this change by the contribution since the projection of data will be modified. With carefully adjusting dimension contributions to each feature, we identified a clear separation (see Figure 5). More specifically, we identified five features as strong dimension contributors that make the Probe and R2L attacks appeared nearby in the PCA projection.

In this project, we introduced a new approach to analyze network traffic data for intrusion detections. Although various approaches have been proposed by incorporating machine learning algorithms, most methods still suffer from detecting unknown attacks. To address the limitation, we proposed a hybrid approach that integrates computational analysis and visual analytics. The computational analysis is used to extract significant features from the data. For visual analytics, an

interactive visualization tool is designed to display the analyzed network traffic features and provide user interaction techniques to support an interactive visual analysis on the visually represented data items. To determine best suitable parameters for applying DWT on the network traffic data, an empirical study was conducted. To show the effectiveness of our approach, the hierarchical cluster method was applied to identify clusters. Although the results indicate that our approach has a strength, it is still important to conduct a formal evaluation study to determine the effectiveness of all possible input and output parameters for DWT and PCA.

Besides the obvious concerns about the ever-growing volume of network traffic data and common trade-off considerations (e.g. overhead vs. speed), there are several major research challenges of utilizing machine learning algorithms in cloud-based network intrusion analysis. There are three salient challenges facing widespread machine learning algorithms implementation in cloud-based network intrusion detection techniques [64]. First, machine learning algorithms trained on a particular dataset may not be suitable for other datasets, and that classification may not be robust over different datasets or domains. Although this remains a critical concern, some researchers have offered preliminary solutions to this problem. Second, machine learning algorithms, in general, are trained using a given number of class types, and hence large varieties of class types found in a dynamically growing dataset could lead to inaccurate classification results. Lastly, machine learning algorithms are developed based on a single learning task, and thus they are not suitable for multiple learning tasks and knowledge transfers required for effective intrusion detection and prevention.

Although Singh et al. [167] provided a concrete example of these latter concerns in our literature, detailing newer botnet architectures that allow for more efficient, less detectable communication inspired by ant-colony foraging behavior, it is important to note that current machine learning techniques cannot uncover or flag these types of instantaneous or stealthy behaviors. Also, we should consider and assume that as newer threats develop, machine learning algorithms will have difficulty remaining one step ahead, without the opportunity to train for newer class types or multiple learning tasks necessary to keep attackers at bay [67].

Even effective classification training cannot provide timely or accurate results for network intrusion. Vieira et al. [166] found that 10 days of usage simulation for Artificial Neural Network training on their intrusion detection techniques fell considerably short, resulting in a high number of false negatives and high uncertainty. The longer a machine learning algorithm takes to complete its learning phase, the slower it will be to adapt to new threats. Moreover, the lack of incremental clustering ability is considered as a possible research challenge [160]. Therefore, feature extraction or reduction algorithms often have to be performed, which could provide considerable overhead and slower response to newer, more sophisticated attacks [94].

Thus, the two-pronged challenge of swifter results and more efficient, nimble training looms large shortly, as intrusion detection techniques work to become more nimble and responsive. In truth, we will likely never see a perfect cloud-based, machine learning algorithm-integrated intrusion detection approach, but some of these deficiencies can be improved with general advancement of the field,

dedicated refinement of algorithms, and some creative problem solving along the way.

8 FUTURE WORKS

For future works, we plan to design a web-based visualization system to analyze network intrusion behaviors. Also, a comparative study with known intrusion detection models will be conducted to determine the benefits and limitations of the system.

Bibliography

[1] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, pp. supl27–supl30, Apr. 2002.

[2] K. Scarfone, K. Scarfone, S. Cybersecurity, P. Mell, R. M. Blank, and A. Secretary, "Guide to intrusion detection and prevention systems (idps)," 2007.

[3] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805–822, 1999.

[4] A. Patcha and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, pp. 3448–3470, Aug. 2007.

[5] L. Halme and R. Bauer, "Aint misbehaving: A taxonomy of anti-intrusion techniques," in *Proceedings of the 18th National Information Systems Security Conference*, 1995.

[6] J. Cannady96 and J. Harrel, "A comparative analysis of current intrusion detection technologies," in *Technology in Information Security Conference (TISC)*, pp. 212–218, 1996.

[7] J. Han and M. Kamber, *Data Mining: Concepts and Techniques: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011.

[8] A. Madhukar and C. Williamson, "A longitudinal study of p2p traffic classification," in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, pp. 179–188, Sept 2006.

[9] M. Dashevskiy and Z. Luo, "Reliable probabilistic classification and its application to internet traffic.," in *ICIC (1)* (D.-S. Huang, D. C. W. II, D. S. Levine, and K.-H. Jo, eds.), vol. 5226 of *Lecture Notes in Computer Science*, pp. 380–388, Springer, 2008.

[10] J.-T. Kim, H.-K. Park, and E.-H. Paik, "Security issues in peer-to-peer systems," in *Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on*, vol. 2, pp. 1059–1063, 2005.

[11] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," in *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, (New York, NY, USA), pp. 512–521, ACM, 2004.

[12] B. Raahemi, W. Zhong, and J. Liu, "Peer-to-peer traffic identification by mining ip layer data streams using concept-adapting very fast decision tree," in *Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on*, vol. 1, pp. 525–532, Nov 2008.

[13] A. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement* (C. Dovrolis, ed.), vol. 3431 of *Lecture Notes in Computer Science*, pp. 41–54, Springer Berlin Heidelberg, 2005.

[14] T. Kushida and Y. Shibata, "Empirical study of inter-arrival packet times and packet losses," in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pp. 233–238, 2002.

[15] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Elsevier Computer Network*, pp. 790–809, 2009.

[16] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy, "Transport layer identification of p2p traffic," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, (New York, NY, USA), pp. 121–134, ACM, 2004.

[17] K. Xu, M. Zhang, M. Ye, D.-M. Chiu, and J. Wu, "Identify p2p traffic by inspecting data transfer behavior.," *Computer Communications*, vol. 33, no. 10, pp. 1141–1150, 2010.

[18] R. Holanda Filho, M. Fontenelle do Carmo, J. Maia, and G. Siqueira, "An internet traffic classification methodology based on statistical discriminators," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pp. 907–910, April 2008.

[19] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 5–16, Oct. 2006.

[20] X. Lu, H. Duan, and X. Li, "Identification of p2p traffic based on the content redistribution characteristic," in *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, pp. 596–601, Oct 2007.

[21] N. Das and T. Sarkar, "Survey on host and network based intrusion detection system," *Int. J. Advanced Networking and Applications*, vol. 6, no. 2, pp. 2266–2269, 2014.

[22] S. Rubin, S. Jha, and B. P. Miller, "Automatic generation and analysis of nids attacks," in *Computer Security Applications Conference, 2004. 20th Annual*, pp. 28–38, IEEE, 2004.

[23] R. Bace, "An introduction to intrusion detection & assessment," *ICSA Intrusion Detection Systems Consortium White Paper*, pp. 1–38, 1999.

[24] S. Kumar and E. H. Spafford, "A pattern matching model for misuse intrusion detection," 1994.

[25] A. Kind, M. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *Network and Service Management, IEEE Transactions on*, vol. 6, pp. 110–121, June 2009.

[26] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Networking, sensing and control, 2004 IEEE international conference on*, vol. 2, pp. 749–754, IEEE, 2004.

[27] K. Worden and J. Dulieu-Barton, "An overview of intelligent fault detection in systems and structures," *Structural Health Monitoring*, vol. 3, no. 1, pp. 85–98, 2004.

[28] G. Duftschmid and S. Miksch, "Knowledge-based verification of clinical guidelines by detection of anomalies," *Artificial intelligence in medicine*, vol. 22, no. 1, pp. 23–41, 2001.

[29] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A neural network based database mining system for credit card fraud detection," in *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pp. 220–226, 1997.

[30] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[31] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*, pp. 77–101, Springer, 2002.

[32] C.-M. Cheng, H. T. Kung, and K.-S. Tan, "Use of spectral analysis in defense against dos attacks.," in *GLOBECOM*, pp. 2143–2148, IEEE, 2002.

[33] H. Wang, D. Zhang, and K. Shin, "Detecting syn flooding attacks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1530–1539, June 2002.

[34] M. Thottan and C. Ji, "Anomaly detection in ip networks," *Signal Processing, IEEE Transactions on*, vol. 51, pp. 2191–2204, Aug 2003.

[35] M. Markou and S. Singh, "Novelty detection: a reviewâĂŤpart 2:: neural network based approaches," *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[36] J. Cannady, "Artificial neural networks for misuse detection," in *National Information Systems Security Conference*, pp. 443–456, 1998.

[37] R. P. Lippmann and R. K. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Computer Networks*, vol. 34, no. 4, pp. 597 – 603, 2000. Recent Advances in Intrusion Detection Systems.

[38] S. T. Sarasamma, Q. A. Zhu, and J. Huff, "Hierarchical kohonenen net for anomaly detection in network security.," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 2, pp. 302–312, 2005.

[39] S.-J. Han, K.-J. Kim, and S.-B. Cho, "Evolutionary learning programÂąÂŕs behavior in neural networks for anomaly detection," in *Neural Information Processing* (N. Pal, N. Kasabov, R. Mudi, S. Pal, and S. Parui, eds.), vol. 3316 of *Lecture Notes in Computer Science*, pp. 236–241, Springer Berlin Heidelberg, 2004.

[40] K. Chadha and S. Jain, "Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks," in *Intelligent Distributed Computing*, pp. 185–198, Springer, 2015.

[41] M. Amini, J. Rezaeenour, and E. Hadavandi, "Effective intrusion detection with a neural network ensemble using fuzzy clustering and stacking combination method," *Journal of Computing and Security*, vol. 1, no. 4, 2015.

[42] J.-H. Lee, J.-H. Lee, S.-G. Sohn, J.-H. Ryu, and T.-M. Chung, "Effective value of decision tree with kdd 99 intrusion detection datasets for intrusion detection system," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, vol. 2, pp. 1170–1175, IEEE, 2008.

[43] C. Kruegel and T. Toth, "Using decision trees to improve signature-based intrusion detection," in *Recent Advances in Intrusion Detection*, pp. 173–191, Springer, 2003.

[44] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with ga-based feature selection," in *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pp. 136–141, ACM, 2005.

[45] R. Jain and N. Abouzakhar, "A comparative study of hidden markov model and support vector machine in anomaly intrusion detection," *Journal of Internet Technology and Secured Transactions (JITST)*, vol. 2, no. 1/2/3/4, pp. 176–184, 2013.

[46] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015.

[47] F. Kuang, S. Zhang, Z. Jin, and W. Xu, "A novel svm by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection," *Soft Computing*, pp. 1–13, 2015.

[48] G. Wang, S. Chen, and J. Liu, "Anomaly-based intrusion detection using multiclass-svm with parameters optimized by pso," 2015.

[49] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. Golkar, and A. Ebrahimi, "A hybrid method consisting of ga and svm for intrusion detection system," *Neural Computing and Applications*, pp. 1–8, 2015.

[50] R. A. Sani and A. Ghasemi, "Learning a new distance metric to improve an svm-clustering based intrusion detection system," in *Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on*, pp. 284–289, IEEE, 2015.

[51] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci.*, vol. 177, pp. 3799–3821, Sept. 2007.

[52] C. Xiang, P. C. Yong, and L. S. Meng, "Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 918–924, 2008.

[53] V. Golmah, "An efficient hybrid intrusion detection system based on c5. 0 and svm," *International Journal of Database Theory and Application*, vol. 7, no. 2, pp. 59–70, 2014.

[54] B. Agarwal and N. Mittal, "Hybrid approach for detection of anomaly network traffic using data mining techniques," *Procedia Technology*, vol. 6, pp. 996–1003, 2012.

[55] X.-s. Gan, J.-s. Duanmu, J.-f. Wang, and W. Cong, "Anomaly intrusion detection based on pls feature extraction and core vector machine," *Knowledge-Based Systems*, vol. 40, pp. 1–6, 2013.

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

[57] W. Yang, C. Sun, and L. Zhang, "A multi-manifold discriminant analysis method for image feature extraction," *Pattern Recognition*, vol. 44, no. 8, pp. 1649–1657, 2011.

[58] S. Sanei, P. Smaragdis, A. T. Ho, A. K. Nandi, and J. Larsen, "Guest editorial: Machine learning for signal processing," *Journal of Signal Processing Systems*, vol. 79, no. 2, pp. 113–116, 2015.

[59] R. Fontugne, J. Mazel, and K. Fukuda, "Hashdoop: A mapreduce framework for network anomaly detection," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pp. 494–499, April 2014.

[60] J. Francois, S. Wang, W. Bronzi, R. State, and T. Engel, "Botcloud: Detecting botnets using mapreduce," in *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*, pp. 1–6, Nov 2011.

[61] M. Kumar and M. Hanumanthappa, "Scalable intrusion detection systems log analysis using cloud computing infrastructure," in *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pp. 1–4, Dec 2013.

[62] Y. Lee and Y. Lee, "Detecting ddos attacks with hadoop," in *Proceedings of The ACM CoNEXT Student Workshop*, CoNEXT '11 Student, (New York, NY, USA), pp. 7:1–7:2, ACM, 2011.

[63] S. Tripathi, B. Gupta, and A. A. A. M. S. Veluru, "Hadoop based defense solution to handle distributed denial of service (ddos) attacks," *Journal of Information Security*, vol. 4, no. 3, pp. 150–164, 2013.

[64] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, pp. 70–73, Apr. 2014.

[65] B. Hu and Y. Shen, "Machine learning based network traffic classification: A survey," *Journal of Information and Computational science*, vol. 9, no. 11, pp. 3161–3170, 2012.

[66] L. Yingqiu, L. Wei, and L. Yunchun, "Network traffic classification using k-means clustering," in *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pp. 360–365, Aug 2007.

[67] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 305–316, IEEE, May 2010.

[68] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[69] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Commun. Surveys Tuts.*, vol. 10, pp. 56–76, Oct. 2008.

[70] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Passive and Active Network Measurement*, pp. 205–214, Springer, 2004.

[71] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.

[72] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on mapreduce," in *Cloud Computing* (M. Jaatun, G. Zhao, and C. Rong, eds.), vol. 5931 of *Lecture Notes in Computer Science*, pp. 674–679, Springer Berlin Heidelberg, 2009.

[73] P. Gupta and N. McKeown, "Algorithms for packet classification," *Network, IEEE*, vol. 15, no. 2, pp. 24–32, 2001.

[74] Y. Qi, L. Xu, B. Yang, Y. Xue, and J. Li, "Packet classification algorithms: From theory to practice," in *INFOCOM 2009, IEEE*, pp. 648–656, April 2009.

[75] J. Erman, A. Mahanti, and M. Arlitt, "Internet traffic identification using machine learning," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pp. 1–6, Nov 2006.

[76] K. Li, C. Gibson, D. Ho, Q. Zhou, J. Kim, O. Buhisi, D. Brown, and M. Gerber, "Assessment of machine learning algorithms in cloud computing frameworks," in *Systems and Information Engineering Design Symposium (SIEDS), 2013 IEEE*, pp. 98–103, April 2013.

[77] K. Singh and S. Agrawal, "Performance evaluation of five machine learning algorithms and three feature selection algorithms for ip traffic classification," *IJCA Special Issue on Evolution in Networks and Computer Communications*, no. 1, pp. 25–32, 2011. Full text available.

[78] M. Stevanovic and J. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 797–801, Feb 2014.

[79] T. Xia, G. Qu, S. Hariri, and M. Yousif, "An efficient network intrusion detection method based on information theory and genetic algorithm," in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pp. 11–17, April 2005.

[80] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Computers & Security*, vol. 24, no. 8, pp. 662 – 674, 2005.

[81] J. Cannady, "Artificial neural networks for misuse detection," in *NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE*, pp. 443–456, 1998.

[82] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, SAC '04, (New York, NY, USA), pp. 420–424, ACM, 2004.

[83] M. Albayati and B. Issac, "Analysis of intelligent classifiers and enhancing the detection accuracy for intrusion detection system," *International Journal of Computational Intelligence Systems*, vol. 8, no. 5, pp. 841–853, 2015.

[84] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal*, vol. 16, pp. 507–521, Oct. 2007.

[85] S. A. Mulay, P. Devale, and G. Garje, "Article:intrusion detection system using support vector machine and decision tree," *International Journal of Computer Applications*, vol. 3, pp. 40–43, June 2010. Published By Foundation of Computer Science.

[86] J. Yao, S. Zhao, and L. Fan, "An enhanced support vector machine model for intrusion detection," in *Proceedings of the First International Conference on Rough Sets and Knowledge Technology*, RSKT'06, (Berlin, Heidelberg), pp. 538–543, Springer-Verlag, 2006.

[87] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Journal of Network and Computer Applications*, vol. 62, pp. 9 – 17, 2016.

[88] N. Kausar, B. Belhaouari Samir, A. Abdullah, I. Ahmad, and M. Hussain, "A review of classification approaches using support vector machine in intrusion detection," in *Informatics Engineering and Information Science: International Conference, ICIEIS 2011, Kuala Lumpur, Malaysia, November 14-16, 2011, Proceedings, Part III* (A. Abd Manaf, S. Sahibuddin, R. Ahmad, S. Mohd Daud, and E. El-Qawasmeh, eds.), (Berlin, Heidelberg), pp. 24–34, Springer Berlin Heidelberg, 2011.

[89] P. G. Majeed and S. Kumar, "Genetic Algorithms in Intrusion Detection Systems: A Survey," *International Journal of Innovation and Applied Studies*, vol. 5, no. 3, pp. 233–240, 2014.

[90] S. N. Pawar and R. S. Bichkar, "Genetic algorithm with variable length chromosomes for network intrusion detection," *International Journal of Automation and Computing*, vol. 12, no. 3, pp. 337–342, 2015.

[91] H. Mohamad Tahir, W. Hasan, A. Md Said, N. H. Zakaria, N. Katuk, N. F. Kabir, M. H. Omar, O. Ghazali, and N. I. Yahya, "Hybrid machine learning technique for intrusion detection system," 5th International Conference on Computing and Informatics (ICOCI) 2015, 2015.

[92] X. Zhang and Q. Wen, "A policy anomaly detecting algorithm based on mapreduce," in *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2011 International Conference on*, vol. 2, pp. 259–262, IEEE, 2011.

[93] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295 – 307, 2005.

[94] S.-Y. Ji, S. Choi, and D. Jeong, "Designing an internet traffic predictive model by applying a signal processing method," *Journal of Network and Systems Management*, pp. 1–18, 2014.

[95] M. Hubert, P. Rousseeuw, and T. Verdonck, "Robust {PCA} for skewed data and its outlier map," *Computational Statistics & Data Analysis*, vol. 53, no. 6, pp. 2264 – 2274, 2009. The Fourth Special Issue on Computational Econometrics.

[96] "Ieee visual analytics challenge 2012, contest chairs- kris cook, georges grinstein, mark whiting." VAST 2012.

[97] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 1313–1329, Aug. 2012.

[98] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," tech. rep., 2005.

[99] "NSL-KDD dataset." `http://nsl.cs.unb.ca/NSL-KDD/`, 2014. [Online; accessed 2-April-2014].

[100] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications*, CISDA'09, (Piscataway, NJ, USA), pp. 53–58, IEEE Press, 2009.

[101] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003.

[102] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, pp. 1263–1284, Sept. 2009.

[103] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, 1st ed., 2013.

[104] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring, "Handling nominal features in anomaly intrusion detection problems," in *Research Issues in Data Engineering: Stream Data Mining and Applications, 2005. RIDE-SDMA 2005. 15th International Workshop on*, pp. 55–62, IEEE, 2005.

[105] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*, vol. 4. Irwin Chicago, 1996.

[106] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman & Hall, 1984.

[107] W.-Y. Loh and N. Vanichsetakul, "Tree-structured classification via generalized discriminant analysis," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. pp. 715–725, 1988.

[108] C. Y. Fu, "Combining loglinear model with classification and regression tree (cart): an application to birth data," *Computational Statistics & Data Analysis*, vol. 45, no. 4, pp. 865 – 874, 2004.

[109] J. Gao, G. Hu, X. Yao, and R. Chang, "Anomaly detection of network traffic based on wavelet packet," in *Communications, 2006. APCC '06. Asia-Pacific Conference on*, pp. 1–5, Aug 2006.

[110] V. Alarcon-Aquino and J. Barria, "Anomaly detection in communication networks using wavelets," *Communications, IEE Proceedings-*, vol. 148, pp. 355–362, Dec 2001.

[111] K. Kyriakopoulos and D. Parish, "Using wavelets for compression and detecting events in anomalous network traffic," in *Systems and Networks Communications, 2009. ICSNC '09. Fourth International Conference on*, pp. 195–200, Sept 2009.

[112] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, (New York, NY, USA), pp. 71–82, ACM, 2002.

[113] A. Dainotti, A. Pescape, and G. Ventre, "Nis04-1: Wavelet-based detection of dos attacks," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pp. 1–6, Nov 2006.

[114] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," in *EURASIP Journal on Advances in Signal Processing*, 2009, (New York, NY, USA), pp. 1–16, Hindawi Publishing Corporation, 2008.

[115] S. Kim, A. Reddy, and M. Vannucci, "Detecting traffic anomalies using discrete wavelet transform," in *Information Networking. Networking Technologies for Broadband and Mobile Networks* (H.-K. Kahng and S. Goto, eds.), vol. 3090 of *Lecture Notes in Computer Science*, pp. 951–961, Springer Berlin Heidelberg, 2004.

[116] C. Callegari, S. Giordano, and M. Pagano, "Application of wavelet packet transform to network anomaly detection," in *Next Generation Teletraffic and Wired/Wireless Advanced Networking* (S. Balandin, D. Moltchanov, and Y. Koucheryavy, eds.), vol. 5174 of *Lecture Notes in Computer Science*, pp. 246–257, Springer Berlin Heidelberg, 2008.

[117] J. Tan, X.-s. Chen, M. Du, and K. Zhu, "A novel internet traffic identification approach using wavelet packet decomposition and neural network," *Journal of Central South University*, vol. 19, no. 8, pp. 2218–2230, 2012.

[118] A. Ramanathan, *WADeS: A Tool for Distributed Denial of Service Attack Detection*. Texas A & M University, 2002.

[119] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proceedings of the IEEE*, vol. 84, pp. 626–638, Apr 1996.

[120] Y. Meyer and R. Ryan, *Wavelets: Algorithms and Applications*. Miscellaneous Bks, Society for Industrial and Applied Mathematics, 1993.

[121] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[122] S. Idicula-Thomas, A. J. Kulkarni, B. D. Kulkarni, V. K. Jayaraman, and P. V. Balaji, "A support vector machine-based method for predicting the propensity of a protein to be soluble or to form inclusion body on overexpression in escherichia coli.," *Bioinformatics*, vol. 22, no. 3, pp. 278–284, 2006.

[123] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[124] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, (London, UK, UK), pp. 137–142, Springer-Verlag, 1998.

[125] J. Hasford, H. Ansari, and K. Lehmann, "Cart and logistic regression analyses of risk factors for first dose hypotension by an ace-inhibitor," *Therapie*, vol. 48, no. 5, pp. 479 – 482, 1993.

[126] P. M. Kuhnert, K.-A. Do, and R. McClure, "Combining non-parametric models with logistic regression: an application to motor vehicle injury data," *Computational Statistics & Data Analysis*, vol. 34, no. 3, pp. 371 – 386, 2000.

[127] W. J. Long, J. L. Griffith, H. P. Selker, and R. B. D'agostino, "A comparison of logistic regression to decision-tree induction in a medical domain," in *Comput. Biomed. Res*, pp. 74–97, 1993.

[128] J. J. Thomas and K. A. Cook, "A visual analytics agenda," *IEEE Comput. Graph. Appl.*, vol. 26, pp. 10–13, Jan. 2006.

[129] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An Interactive System for PCA-based Visual Analytics," *Computer Graphics Forum*, 2009.

[130] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, pp. 50–61, June 1995.

[131] S.-Y. Ji, S. Choi, and D. H. Jeong, "Designing an internet traffic predictive model by applying a signal processing method," *Journal of Network and Systems Management*, vol. 23, no. 4, pp. 998–1015, 2015.

[132] B. AsSadhan, H. Kim, J. M. F. Moura, and X. Wang, "Network traffic behavior analysis by decomposition into control and data planes," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pp. 1–8, April 2008.

[133] D. Jiang, J. Liu, Z. Xu, and W. Qin, "Network traffic anomaly detection based on sliding window," in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, pp. 4830–4833, Sept 2011.

[134] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, (New York, NY, USA), pp. 71–82, ACM, 2002.

[135] C. T. Huang, S. Thareja, and Y. J. Shin, "Wavelet-based real time detection of network traffic anomalies," in *Securecomm and Workshops, 2006*, pp. 1–7, Aug 2006.

[136] S. S. Kim, A. L. N. Reddy, and M. Vannucci, *Detecting Traffic Anomalies Using Discrete Wavelet Transform*, pp. 951–961. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[137] G. Liu, Z. Yi, and S. Yang, "A hierarchical intrusion detection model based on the pca neural networks," *Neurocomputing*, vol. 70, no. 7–9, pp. 1561 – 1568, 2007. Advances in Computational Intelligence and Learning14th European Symposium on Artificial Neural Networks 200614th European Symposium on Artificial Neural Networks 2006.

[138] Y. Bouzida, "Intrusion detection using principal component analysis," in *In Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, 2003.

[139] G. E. Plassman, "A survey of singular value decomposition methods and performance comparison of some available serial codes." NASA Technical Report CR-2005-213500, October 2005.

[140] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20–30, 2006.

[141] I. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, Springer, 2002.

[142] Y.-F. Sang, D. Wang, and J.-C. Wu, "Entropy-based method of choosing the decomposition level in wavelet threshold de-noising," *Entropy*, vol. 12, no. 6, p. 1499, 2010.

[143] L. Monzón, G. Beylkin, and W. Hereman, "Compactly supported wavelets based on almost interpolating and nearly linear phase filters (coiflets)," *Applied and Computational Harmonic Analysis*, vol. 7, no. 2, pp. 184 – 210, 1999.

[144] B. Schölkopf, J. Platt, and T. Hofmann, *In-Network PCA and Anomaly Detection*, pp. 617–624. MIT Press, 2007.

[145] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 3, pp. 311–336, 2011.

[146] R. M. Esteves, R. Pais, and C. Rong, "K-means clustering in the cloud – a mahout test," in *Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, WAINA '11, (Washington, DC, USA), pp. 514–519, IEEE Computer Society, 2011.

[147] Z. Chen, F. Han, J. Cao, X. Jiang, and S. Chen, "Cloud computing-based forensic analysis for collaborative network security management system," *Tsinghua Science and Technology*, vol. 18, pp. 40–50, Feb 2013.

[148] S. Muthurajkumar, K. Kulothungan, M. Vijayalakshmi, N. Jaisankar, and A. Kannan, "A rough set based feature selection algorithm for effective intrusion detection in cloud model," 2013.

[149] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action.* Greenwich, CT, USA: Manning Publications Co., 2011.

[150] X. Meng, J. K. Bradley, B. Yavuz, E. R. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "Mllib: Machine learning in apache spark," *CoRR*, vol. abs/1505.06807, 2015.

[151] A. Ghoting, R. Krishnamurthy, E. Pednault, B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian, and S. Vaithyanathan, "Systemml: Declarative machine learning on mapreduce," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, (Washington, DC, USA), pp. 231–242, IEEE Computer Society, 2011.

[152] A. Ghoting, P. Kambadur, E. Pednault, and R. Kannan, "Nimble: A toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, (New York, NY, USA), pp. 334–342, ACM, 2011.

[153] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan, "Mlbase: A distributed machine-learning system.," in *CIDR*, www.cidrdb.org, 2013.

[154] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," *Proc. VLDB Endow.*, vol. 5, pp. 716–727, Apr. 2012.

[155] A. G. Andrew Crotty and T. Kraska, "Tupleware: Distributed machine learning on small clusters," *IEEE Data Eng. Bull.*, vol. 37, no. 3, pp. 63–76, 2014.

[156] M. Boehm, A. V. Evfimievski, N. Pansare, and B. Reinwald, "Declarative machine learning - A classification of basic properties and types," *CoRR*, vol. abs/1605.05826, 2016.

[157] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, pp. 567–581, Mar. 2013.

[158] Y. Lee, W. Kang, and H. Son, "An internet traffic analysis method with mapreduce," in *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pp. 357–361, April 2010.

[159] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," in *ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference*, (New York, NY, USA), pp. 136–141, ACM, 2005.

[160] T. Chen, X. Zhang, S. Jin, and O. Kim, "Efficient classification using parallel and scalable compressed model and its application on intrusion detection," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5972–5983, 2014.

[161] X. Shu, J. Smiy, D. D. Yao, and H. Lin, "Massive distributed and parallel log analysis for organizational security," in *2013 IEEE Globecom Workshops (GC Wkshps)*, pp. 194–199, Dec 2013.

[162] Y. Zhai, Y. S. Ong, and I. W. Tsang, "The emerging "big dimensionality"," *IEEE Computational Intelligence Magazine*, vol. 9, pp. 14–26, Aug 2014.

[163] I. K. Hyunjoo Kim, Jonghyun Kim and T. myung Chung, "Behavior-based anomaly detection on big data," in *Proceedings of 13th Australian Information Security Management Conference*, pp. 73–80, 2015.

[164] M. K. Amy Xuyang Tan, Valerie Li Liu and B. Thuraisingham, "A comparison of approaches for large-scale data mining," Tech. Rep. UTDSC-24-10, University of Texas at Dallas, Department of Computer Science, 2010.

[165] I. Aljarah and S. A. Ludwig, "Mapreduce intrusion detection system based on a particle swarm optimization clustering algorithm," in *2013 IEEE Congress on Evolutionary Computation*, pp. 955–962, June 2013.

[166] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," *It Professional*, no. 4, pp. 38–43, 2009.

[167] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Information Sciences*, vol. 278, no. 0, pp. 488 – 497, 2014.

[168] A. H. Bhat, S. Patra, and D. Jena, "Machine learning approach for intrusion detection on cloud virtual machines," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 6, pp. 56–66, 2013.

[169] A. Marnerides, M. Watson, N. Shirazi, A. Mauthe, and D. Hutchison, "Malware analysis in cloud computing: Network and system characteristics," in *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp. 482–487, Dec 2013.

[170] H. Wang, W. Ding, and Z. Xia, "A cloud-pattern based network traffic analysis platform for passive measurement," in *Cloud and Service Computing (CSC), 2012 International Conference on*, pp. 1–7, Nov 2012.