



ANOMALOUS NETWORK PACKET DETECTION

AHMED BURHAN MOHAMMED

MAY 2015

ANOMALOUS NETWORK PACKET DETECTION

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY**

**BY
AHMED BURHAN MOHAMMED**

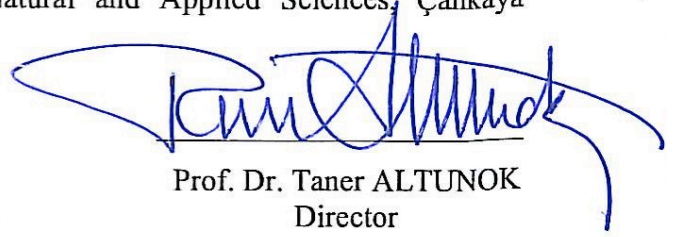
**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING**

MAY 2015

Title of the Thesis: **Anomalous Network Packet Detection.**

Submitted by **Ahmed Burhan MOHAMMED.**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.



Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Assist. Prof. Dr. Tansel ÖZYER
Co-Supervisor



Assist. Prof. Dr. Sibel TARIYAN ÖZYER
Supervisor

Examination Date: 08.05.2015

Examining Committee Members

Assist. Prof. Dr. Sibel TARIYAN ÖZYER

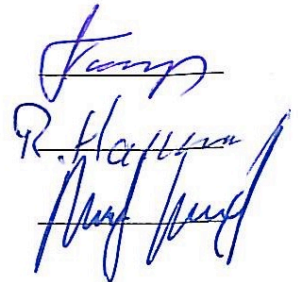
(Çankaya Univ.)

Assist. Prof. Dr. Reza ZARE HASSANPOUR

(Çankaya Univ.)

Assoc. Prof. Dr. Murat KOYUNCU


(Atılım Univ.)



STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Ahmed, MOHAMMED

Signature : 

Date : 08.05.2015

ABSTRACT

ANOMALOUS NETWORK PACKET DETECTION

MOHAMMED, Ahmed Burhan

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Sibel TARIYAN ÖZYER

May 2015, 53 pages

In the last decade, extensive research has been done to the improvement of Intrusion Detection Systems (IDS) for anomalous network packets. Two types of IDS are available. The first one is the signature-based detection system. It can detect intrusions by scanning network packets and compare them with human-generated signatures against previously observed attacks. The second type is the anomaly-based detection system, which is able to detect new attacks against observed attacks. In this thesis, anomaly-based detection systems have been used with density base clustering algorithms and techniques. DBSCAN (Density-Based Spatial Clustering and Application with Noise) and DenStream algorithms are well-known data stream clustering algorithms in data mining. DBSCAN algorithm can separate packets as normal and noisy data. The second algorithm, DenStream starts off with DBSCAN and then tries to reduce the amount of noise to be clustered. For this study, we used DARPA' 99 dataset. We worked with attacks of type R2L, U2R, DoS and Probe. The DenStream and DBSCAN algorithms have been performed with fine-tuned. Overall, the DenStream algorithm achieved higher detection results and sensitivity

than the DBSCAN algorithm. After, only epsilon distance and minimum number of points parameters for neighborhood area are fine-tuned, the clustering methods can be easily applied for classifying normal and noisy data regardless of its attack type.

Keywords: Anomaly Detection, Clustering, Data Stream Mining, DBSCAN, DenStream and Intrusion Detection System.

ÖZ

ANOMAL AĞ PAKETİ TESPİTİ

MOHAMMED, Ahmed Burhan

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Sibel TARIYAN ÖZYER

Mayıs 2015, 53 sayfa

Geçen on yıllık dönemde anomal ağ paketlerinin Saldırı Tespit Sistemlerini (STS) iyileştirmek amacıyla yoğun araştırmalar yapılmıştır. İki türden STS mevcuttur. Birincisi imzaya dayalı tespit sistemidir. Bu sistem, ağ paketlerini tarayarak ve bunları önceden gözlemlenen saldırılara karşı, insan kaynaklı imzalarla karşılaştırarak saldırıları tespit edebilmektedir. İkinci tip, gözlemlenen ataklara karşı yeni saldırıları tespit edebilen, anomaliye dayalı tespit sistemidir. Bu tezde, anomaliye dayalı tespit sistemi, yoğunluk tabanlı kümeleme algoritma ve teknikleri ile kullanılmıştır. DBSCAN (Yoğunluk Tabanlı Uzaysal Kümeleme ve Gürültülü Veride Uygulaması) ve DenStream algoritmaları veri madenciliğinde iyi bilinen yoğunluk kümeleme algoritmalarıdır. DBSCAN algoritması, paketleri normal ve

vi

gürültülü veri olarak ayrıştırabilir; ikinci algoritma, DenStream, DBSCAN ile başlar ve sonra kümelenecek gürültü miktarını azaltmaya çalışır. Bu çalışma için DARPA'99 veri seti kullanılmıştır. R2L, U2R, DoS ve Probe atak tipleri ile çalışılmıştır. DBSCAN ve DenStream algoritmaları ince ayar ile uygulanmıştır. Performans ve etkinliğe göre, DenStream algoritması, DBSCAN algoritmasından daha yüksek tespit sonuçları ve hassasiyet elde etmiştir. Daha sonra yalnızca epsilon uzaklık ve en az nokta sayısı parametreleri komşu alan için ince ayarlanarak; kümeleme yöntemleri, saldırı tipinden bağımsız, normal ve gürültülü veri sınıflandırma için kolaylıkla uygulanabilmektedir.

Anahtar kelimeleri: Anomali Tespiti, Kümelene, Veri Akışı Madenciliği, DBSCAN, DenStream ve Saldırı Tespit Sistemi.

ACKNOWLEDGEMENTS

I would like to gratefully and sincerely thank Assist. Prof. Dr. Sibel TARIYAN ÖZYER, for her guidance, understanding, patience, and most importantly, her friendship during my graduate studies.

I would like to seriously thank Assist. Prof. Dr. Tansel ÖZYER for his guidance, understanding, helps and recommends.

I would like to thank Prof. Dr. Taner ALTUNOK who is the head of Graduate School of Natural and Applied Sciences, Çankaya University.

I would also like to give special thanks to Assist. Prof. Dr. Najdat DAMERCI from Kirkuk University for his help to complete my graduate study. Also, I would like to thank Assist. Prof. Nooraldeen Ibrahim Abdulah.

Wholeheartedly, I would like to thank my parents, Burhan and Ramziyah, their love and trust in me. I gained ability to tackle challenges with their watchful eyes.

Most importantly, I would like to thank my wife Aya. Her support, her encouragement, her quiet patience and her unwavering love were undeniable in the past eight years of my life. Thanks to her tolerance devotion and love. Also thanks to my daughters (Wldan and Wijdan).

Finally, I would like to thanks, everyone who help me during my study.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF FIGURES.....	xi
LIST OF TABLES.....	xiii
LIST OF ABBREVIATIONS.....	xiv

CHAPTERS:

1. INTRODUCTION.....	1
1.1. Context.....	1
1.2. Related Work	2
1.3. Objectives.....	6
1.4. Structure of the Thesis	6
2. BACKGROUND.....	7
2.1. Data Mining.....	7
2.1.1. Steps of data mining:.....	8
2.1.2. Data mining main topics.....	9
2.1.3. Architecture of a normal data mining system.....	10
2.2. Data Stream Mining.....	10
2.3. Clustering.....	11
2.3.1. Classical requirements of clustering.....	12
2.3.2. Main clustering methods categories:.....	13
2.4. Cyber Attacks Types.....	15
2.4.1. Attack types.....	15
2.4.2. Network protocols attacks.....	16

2.4.3.	Malicious code types.....	19
2.5.	Intrusion Detection Systems.....	20
2.5.1.	Characteristic approaches.....	20
2.5.2.	IDS classifications approaches.....	21
2.5.3.	Detection approaches.....	22
3.	DATASET AND METHOD.....	25
3.1.	DARPA Dataset.....	25
3.2.	DBSCAN Algorithm.....	26
3.3.	DenStream Algorithm.....	27
3.4.	Sensitivity and Specificity.....	29
4.	EXPERIMENTS AND RESULTS.....	32
4.1.	Experiments of DBSCAN Algorithm.....	32
4.2.	Experiments of DenStream Algorithm.....	38
4.3.	Experiments Over 2000 Packets.....	44
5.	DISCUSSION.....	46
5.1.	F1 Score for DBSCAN and DenStream.....	46
5.2.	Sensitivity of DBSCAN and DenStream.....	47
5.3.	FPR for DBSCAN and DenStream.....	47
5.4.	Attack Detection for DBSCAN and DenStream.....	48
5.5.	Running Time to Get Result.....	49
6.	CONCLUSION AND FUTURE WORK.....	52
	REFERENCES.....	R1
	APPENDICES.....	A1
A.	CURRICULUM VITAE.....	A1

LIST OF FIGURES

FIGURES

Figure 1	Data mining steps.....	8
Figure 2	Partition based method example.....	13
Figure 3	Hierarchical based method example.....	13
Figure 4	Density based method example.....	14
Figure 5	Grid based method example.....	14
Figure 6	Cyber intrusion and attacks.....	20
Figure 7	IDS component.....	21
Figure 8	IDS categorization.....	24
Figure 9	2X2 Conditions and test outcome.....	30
Figure 10	DBSCAN procedure.....	32
Figure 11	F1 score for DBSCAN.....	33
Figure 12	Output result for F1 with attacks.....	35
Figure 13	Sensitivity of DBSCAN.....	36
Figure 14	Attack detected with DBSCAN.....	36
Figure 15	TPR and FPR of DBSCAN.....	37
Figure 16	DenStream procedure.....	38
Figure 17	F1 outputs in DenStream.....	39
Figure 18	Output result for F1 with attacks.....	41
Figure 19	Sensitivity of DenStream.....	42
Figure 20	Attacks detected with DenStream.....	42
Figure 21	TPR and FPR of DenStream.....	43
Figure 22	F1 score result for DBSCAN.....	44
Figure 23	F1 score result for DenStream.....	45
Figure 24	F1 score for both algorithms.....	46

FIGURES

Figure 25	TPR for both algorithms.....	47
Figure 26	FPR for both algorithms.....	48
Figure 27	Number of attacks detected for both algorithms.....	49
Figure 28	Running time for both algorithms.....	49
Figure 29	Running time comparison.....	51

LIST OF TABLES

TABLES

Table 1	Highest Output Result of F1.....	33
Table 2	Cases of DBSCAN.	34
Table 3	Output Results of F1 with Attacks	34
Table 4	Output Result of TPR.....	35
Table 5	TPR and FPR of DBSCAN.....	37
Table 6	Duration Process Time	38
Table 7	Highest Output Result of F1.....	40
Table 8	Cases of DenStream.....	40
Table 9	Output Result of F1.....	41
Table 10	Output Result of TPR.....	41
Table 11	Output Result of TPR and FPR.....	43
Table 12	Detection Process Time.....	50
Table 13	Running Time Both Algorithms.....	50
Table 14	Sensitivity Higher Results.....	52

LIST OF ABBREVIATIONS

IDS	Intrusion Detection System
NIDES	Next-generation Intrusion Detection Expert System
PHAD	Packet Header Anomaly Detection
ALAD	Application Layer Anomaly Detection
NETAD	Network Traffic Anomaly Detection
PAYL	Payload-based Anomaly Detector for Intrusion Detection
LCS	Longest Common Substring
McPAD	Multiple Classifier Payload-based Anomaly Detector
SVM	Support Vector Machine
HMM	Hidden Markov Models
DBSCAN	Density-Based Spatial Clustering and Application with Noise
DM	Data Mining
KD	Knowledge Discovery
KDD	Knowledge Discovery Database
DSM	Data Stream Mining
ICMP	Internet Control Message Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
MAC	Media Access Control
ARP	Address Resolution Protocol
UPD	User Datagram Protocol
DNS	Domain Name System
SMTP	Simple Mail Transfer Protocol
URL	Uniform Resource Locator
NIDS	Network Intrusion Detection System
HIDS	Host Intrusion Detection System

ACK	Acknowledgement
SYN	Synchronize
DARPA	Defense Advanced Research Projects Agency
R2L	Remote to Local
U2R	User to Root
DoS	Denial of Service
(<i>e</i>)	Epsilon
(<i>mp</i>)	Minimum Point
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
TPR	True Positive Rate
FPR	False Positive Rate

CHAPTER 1

INTRODUCTION

1.1 Context

In the 21st century, the internet has become a vital part of our life [1], and it is very difficult to live on this planet without using it frequently. Moreover, the network has become a universal protocols for a lot of activities. Social networking, credit card, file sharing, payment processing, streaming media, financial information, e-mail addresses, login passwords and other web online applications [2]. Some off users write these applications and have 25% vulnerabilities out of total network security [1].

In the previous two decades, the internet was limited to desktop or laptop computers. However, today after the great explosion of mobile technology with new generations and Android systems, new developments have made using the internet very easy [3]. However, these developments increased the number of internet users and also potential victims compared to the previous decade [4].

Life is dependent on good and evil, and the internet is also like this. We can use it for good but there are a lot of bad things, and people use it on the other side. Daily there is a contest between good (helpful users) and bad (hackers). Everyday programmers develop new software to use vulnerabilities of web services to attack users and to propagate “viruses, malicious files, spyware, worms, and trojans. These types of attacks lead attackers to destroy files, stealing personal data, capturing passwords, etc.” [2].

According to a “Symantec Internet security threat”[4] report in 2014. Gives the 2013-year name “Mega Breach”[2], after giving the 2012-year name a “Data

Breach”[4]. The total number of breaches in 2013 was 62 percent greater than in 2012 with 253 total violations [3]. It was also larger than the 208 breaches in 2011. In addition, web attackers blocked per day “190,000” in 2011, “464,100” in 2012 and “568,700” in 2013. This mean HTTP most common protocol under attack although 3 of top ten types of information breaches was “Real Name, Birth Date and Government ID number” [2].

More Zero-day vulnerabilities discovered in 2013 than any other year Symantec has tracked. The 23 zero-day vulnerabilities discovered represent a 62 percent increase over 2012 and are more than the two previous years combined [2]. Zero-day vulnerabilities are coveted because they give attackers a way to infect their victim silently. Unfortunately, 77 percent of legitimate websites had exploitable vulnerabilities and 1-in-8 of all websites had a critical vulnerability [3].

Gives attackers plenty of choices in websites to place their malware and entrap their victims, besides which vulnerabilities within internet protocols is used by attackers to detect attackers files. Network admin needs more programs and software with hardware for an Intrusion Detection System (IDS) [5].

The internet is growing faster. So detecting attackers is more complex. Moreover, challenging attacks needs a new type of network traffic analysis to hold big data with fast analytic capabilities [1], so Machine Learning algorithms are used to work on network attackers detection. They were armed with intrusion detection systems based on the internal similarity between instances and clustering with K-Means, DBSCAN and CLARANS [6].

1.2 Related Work

Intrusion Detection System is one notable issue of network security research to develop, which attempts to detect threats, malicious files, and abnormal attackers that threaten the network or the host [7]. There are some reasons that make IDS be a necessary part of our network for the entire defense system [8].

Firstly, many traditional systems, applications and network protocols were designed without security in mind. Secondly, some network protocols have vulnerabilities in which developing IDS needs more work and expenditure to strengthen defense threads. IDS can divide into two broad categories: approaches that are dependent on knowledge-based or signature-based “misuse” and behavior based or learning-based “anomaly” [9]. Network intrusion detection systems inspect the network traffic in real time mode and initiate security alarms when potentially malicious data is detected [10].

In this section, we discuss some examples of related researchers with network packet detection. Misuse is a rule-based network IDS such as Bro and Snort, but these types cannot do more to prevent zero-day worms [11]. They are designed with an open signature to detect well-known types after a worm has been launched successfully.

The misuse signature-based IDS is a technique that contains an amount of attack descriptions or signatures that are matches against all streams of review data watching for evidence of modeled attack [12]. This method can used for previously known attack detection, and the outline of the attacker has to be manually revised. So anonymous attacks in pattern signature cannot capture. Misuse based detection can find and detect known attacks successfully [13].

Anomaly-based detection is a technique that can effectively detect new attacks, but this technique is more complex than previous ones and is very costly and raises a lot of false alarms. Anomaly-based IDS adaptively detect new attacks by first generating a “normal” pattern of network traffic [10]. Then they find anomalous packets by comparing incoming packets with the normal packet model [11].

In this section, we present some systems of anomaly base detection:

NIDES (Next-generation Intrusion Detection Expert System): This system builds a model for detecting an anomaly by behavior monitoring the first four tuples of the packet. It has to model the destination and source IP address and port numbers of the packet header [8].

PHAD (Packet Header Anomaly Detection): This system monitors 33 fields in a single packet, which features IP (Internet Protocol), Ethernet and Transport layers in a packet header (TCP, UDP, ICMP) [13].

ALAD (Application Layer Anomaly Detection): This model monitors the incoming server TCP connection. It features application protocol keyword, open and close TCP flag, destination and source addresses and port number, but it is not like PHAD [14].

NETAD (Network Traffic Anomaly Detection): Is another detection model using the network packet header information. It examines the first 48 bytes from each IP packet and creates a different model for each corresponding to a network protocol [8].

PAYL (Payload-based Anomaly Detector for Intrusion Detection): This method based on simple statistics extracted from the payload [12]. Utilizes 1-gram it is fully hands-free online anomaly detection, uses Longest Common Substring (LCS) [1]. It is self-calibrating automatically observes itself and update models. The traditional version was with one center, but most new features are implemented to use multiple centroids and ingress/egress correlation [11].

McPAD (Multiple Classifier Payload-based Anomaly Detector): This is a model n-gram version more developed than PAYL. Using 2-gram with the combination of multiple one classes of Support Vector Machine (SVM) [12] classifiers to detect anomaly packets. McPAD is right to detect new virus types if there is a false positive rate low [1].

All previous model types to detect anomaly packets have been unable to store and evaluate the significant amount of network traffic. Many researchers use Machine Learning and Data Stream Mining algorithms to counter this problem.

Finally, I present four papers featuring sample works on anomaly Intrusion Detection Systems using Data Mining Algorithms and Techniques:

Miller Z., Deitrick W., Hu W.: using data stream mining algorithms, with an n-gram packet payload to anomalous IDS. Using “DARPA’99” dataset in “ARFF” file format to work on it in open source machine learning application “WEKA” [1]. Besides, presents two types of algorithms: first is the clustering anomaly base detection with DenStream algorithm developed for detecting HTTP attacks according to the attack with Generic HTTP, Shellcode and Polymorphic attacks inserted. Also, the second is a histogram anomaly-based detection that creates a model for known normal packet payload and compares it with incoming packets [1].

Miller Z., Hu W.: using modified the density-based algorithm for clustering-based anomaly detection within clustering “preDeCon” that is inspired by DBSCAN and it is of the generation Optics. To work on high-dimensional data without decreasing the accuracy of clustering models we use “preDeCon” algorithm for a nation of preference subspace in clustering work with DARPA dataset [6].

Oza A., Ross K., Low R. M., Stamp M.: using three types of analyzing and techniques on bytes-based analysis for constructing models of benign (X2 distance, Ad-hoc n-gram distance and pattern distance). Depend on n-gram method to extract features. HTTP attacks using DARPA data with machine learning algorithms Hidden Markov Models (HMM) to detect anomaly and classify it as benign or malicious [10].

Faizal M. A., Mohd Z. M., Sahib S.: using time based detection a novel approach for detecting fast attack as an intrusion detection system, author use Bro system to generate alarm when capture the attacker IP address that can be identify [9]. In addition, time based model using IP address with specific time for searching and comparing each IP address based time interval to classify this table paper use logistic regression model to classification table based on detection attack, “Null model and Full model”[9], the logistic regression will produce to identify the accuracy of detection.

1.3 Objectives

In this thesis, the researcher attempts to detect anomalous network packets. Using the data mining techniques “Stream Mining and Clustering”. By applying 1-gram method on the packets of network traffic. Working on dataset “DARPA’99”. By implementing of two density base algorithms. DBSCAN (Density-Based Spatial Clustering and Application with Noise) algorithm and DenStream (Density Stream) algorithm. Then comparing the output results for both algorithms. According to the sensitivity and specificity formula. Our system differs in two ways from [1]. Our system can run on random subsamples of the entire dataset. Besides, the best result is achieved by doing a grid search for the best parameter set. This approach used different attack categories.

1.4 Structure of the Thesis

The structure of this thesis is prepared as follows, with six chapters that cover information about anomalous network packet detection:

Chapter 1. Presents context and related works in anomaly detection with machine learning.

Chapter 2. Prepared in two parts with definitions. Firstly, it describes data mining with stream mining techniques and clustering. Secondly, it describes network attacks and intrusion detection system.

Chapter 3. Presents dataset with two density base algorithms DBSCAN and DenStream with thesis work procedures.

Chapter 4. Prepared by experimentation of producers and algorithms used in this thesis.

Chapter 5. Presents discussion of the output results collected from experiments. Finally, Chapter 6 presents the conclusion and future work.

CHAPTER 2

BACKGROUND

2.1 Data Mining

Data Mining (DM) and Knowledge Discovery (KD): Data Mining has involved a lot of attention to knowledge activity in recent years, due to the varied availability of huge amounts of data and the consequent need to know the types of data that are relevant. [15].

DM is defined as the process of determining patterns in data. The process of determining can be manual or dynamical. The patterns learned must be relevant and lead to some advantage [16].

Data mining can be considered as a result of the normal development of information technology. Also, DM refers to gaining or “mining” information from huge amounts of data [15]. Mining is an expressive phrase illustrating the process that catches a trivial set of valuable pieces from a huge lode of sensitive material. Other definitions of data mining could be “knowledge mining from data, knowledge extraction, data analysis, data archaeology” [17], and searching for data.

Another understanding of data mining is Knowledge Discovery in Databases, or KDD.

Knowledge Discovery in Databases (KDD): is an automatic, investigative large data repository for analyzing and modeling. KDD is the prepared access of detecting valuable, legal, original, and logical patterns from huge, composite data sets. Data Mining (DM) is the essence of the KDD process, including the gathering of algorithms that investigate the data, improve the model and determine formerly unidentified patterns [18].

Data mining has many different perspectives “Statistics, Artificial intelligence, Database management systems, Pattern recognition”. Additionally it has different rules and methods: Generalization, Classification, Association, Clustering, Stream data mining, Intrusion detection, Time-series analysis, “Frequent pattern analysis, Structured pattern analysis, Trend analysis, Outlier analysis, Biological data mining, Deviation analysis, Web mining, Text mining,” [19], Privacy-preserving mining.

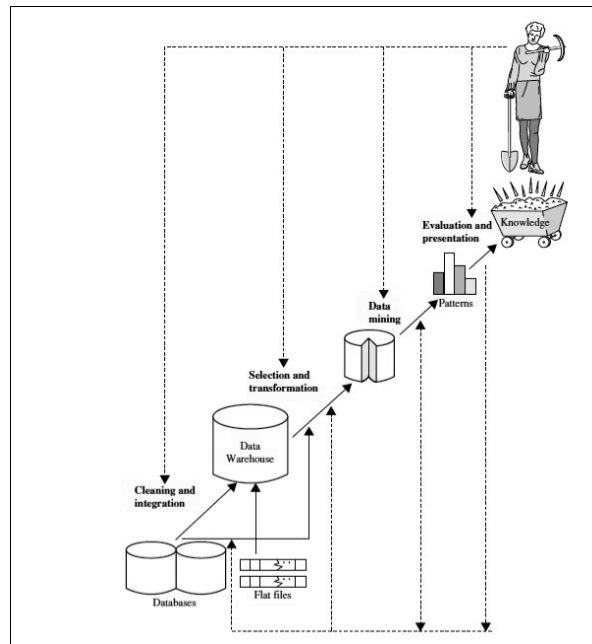


Figure 1 Data mining steps [15]

2.1.1 Steps of data mining

As shown in Fig.1 data mining has several steps:

- Data cleaning: (to subtract noise).
- Data integration: (multiple data may be merged).
- Data selection: (related data and study database tasks).
- Data transformation: (transformed data into forms suitable for mining, by execution aggregation processes for instance) [15].
- Data mining: (intelligent approaches use to obtain data forms).
- Pattern evaluation: (classify the truly stimulating patterns indicating knowledge).
- Knowledge presentation: (imagining and knowledge depiction methods are deployed to display the mined knowledge to the user) [17].

2.1.2 Data mining main topics

The scopes of many data mining books run to major issues in data mining regarding mining methodology, user interaction, performance, and diverse data types [17]. These issues are introduced below [18]:

- Mining methodology and user interaction issues: These reflect the kinds of knowledge mined, the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad hoc mining, and knowledge visualization.
- Mining different kinds of knowledge in databases: Data mining includes different type of methods and algorithms so any user can obtain different knowledge from same data set by using more analysis techniques.
- Interactive mining of knowledge at multiple levels of abstraction: how to help the user focus on the search to find knowledge on request, with sampling techniques, which are results, focused. Additionally, the user can discover patterns at different granularities from varying perspectives.
- Incorporation of background knowledge: this is knowledge concerning the area covered by the discovery processes to discover patterns, related to databases.
- Data mining query languages: users can use many type of queries such as (SQL) to find and develop analysis rules for mining knowledge.
- Presentation and visualization of data mining results: to view the result as using visualizing methods in data mining can represent figures.
- Handling noisy or incomplete data: noise can yield distorted results using data mining techniques. Clean data sets are essential in avoiding these abnormalities.
- Pattern evaluation not motivating problem: some patterns are of no interest to the user so user-specified checks monitor the discovery process and decrease the search space.
- Performance issues: refer to effectiveness, scalability and parallelization of data mining algorithms.
- Efficiency and scalability of DM algorithms: working with running time of DM algorithms on data set must be expectable and suitable in huge databases.
- Parallel, distributed, and incremental mining algorithms: massive data sets and repositories are stored in different types and so are distributed in many places. Data

mining includes parallel process algorithms to analyze this type of massively distributed data sets.

2.1.3 Architecture of a normal data mining system

- Information Repository: refers to a group of data repositories or many kinds of information. The data must be clean and have data integration.
- Database or data warehouse server: refers to kind of servers searching for related data based on user request to mining.
- Knowledge base: refers to the knowledge area used to find or calculate the interest emanating from patterns.
- Data mining engine: refers to a set of function modules such as classification, clustering, and evolution analysis.
- Pattern evaluation module: refers to the pattern valuation module. This may be integrated with the mining module, and be conditional on the operation of the data mining.
- User interface: refers to the links between users and the data mining system and allows the user to (look at database, calculate mined patterns and visualize the results).

2.2 Data Stream Mining

Data Stream Mining (DSM): can be defined as a part of data mining, and Machine Learning [20]. DSM is a method of obtaining information and knowledge from continuous fast data, streamed real time. This is a well-organized series of illustrations that in many applications of DSM can be read in real time. Such data streams usually cannot save in any kind of data store [21].

Data streams are unbounded data arriving in strings. The domain of probable values can be large for an attribute. Data streams have specific features: huge or unbounded in size or infinite volume; data can change dynamically, data rolling in and out in a stable order so the system cannot control the stream data order; a small number of stream data allow for scan, and the stream data work on line (in fast real time)

respond to time [21]. Examples of data streams could be “computer network traffic, mobile exchanges data, stock exchange and ATM transactions, search engine on web, power supply, video following, weather or environment monitoring, and sensor network data” [22].

The goal of stream mining is to calculate the value or class of new instances in the data stream, assuming some knowledge of class membership giving previous instances values [16]. To learn the calculation task from labeled examples one can use Machine Learning techniques. The goal of the calculation, the class that is to be the target value or else the predicted class, may change over time. So active management of stream data analysis presents challenges to researchers [23].

Mining data streams includes the effective detection of general patterns and dynamic changes within stream data. In computer network we use it for detect intrusions based on the anomaly of message flow, which may be learned by clustering algorithms of data streams. Thus, in many different levels and multiple dimensional systems analysis and online analysis and mining should be implemented on stream data as well.

2.3 Clustering

Clustering: Is the separation of a set of physical or abstract objects into groups, or clusters, so that items are “similar” within a cluster and are “dissimilar” to objects in other clusters [24]. Similarity is normally defined as a relationship between objects and how “close” the objects are in space, dependent on a distance function. The “quality” of a cluster may be denoted by the cluster’s diameter. The centroid distance is another aspect of cluster quality and is defined as the typical distance of any cluster object from the cluster [25].

Another meaning of clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters [21]. Dissimilarities and similarities are assessed, based on the attribute values describing the objects and often involve

distance measurements. Clustering as a data-mining tool has its roots in many application areas such as biology, security, business intelligence, and Web search [16]. Clustering analysis commonly used in numerous applications, involving “data analysis, pattern recognition and image processing”. Clustering can also be: data segmentation in some applications because clustering separations large data sets into groups depending on their similarity. It takes into account the efficiency of clustering methods for composite kinds and forms of data, and high level clustering with mixed numerical and unqualified data [26].

2.3.1 Classical requirements of clustering

- Scalability: Several clustering algorithms work with small data as well which contain several data objects. For a large data set we need high scalable algorithms for clustering.
- Discovery of clusters with random shape: Several clustering algorithms resolve clusters by calculating distance measurements: these Algorithms tend to find spherical clusters with similar size and density [24].
- Determine input parameters by the smallest supplies for a given knowledge area: Several clustering algorithms involve input clustering analysis within assured parameters: the results can be sensitive to input parameters.
- Agreement with noisy data: Several clustering algorithms are sensitive to noise: in addition, most data sets in the real world contain missing data or unknown input data that makes for poor clustering [25].
- Insensitivity clustering to new input records: Several clustering algorithms cannot deal with new input records to the database: there is a need to develop clustering methods to determine new clusters that are incremental and sensitive to new data.
- High dimensionality: Some data in data repositories have more than two or three dimensions or attributes so several clustering algorithms are not good to have more data dimensions then finding cluster to work in this type of data become challenge to do.

2.3.2 Main clustering methods categories

A- Partitioning methods: These methods focus on instances in which 1-level separates data sets. Each object must be in one group precisely [24] as shown in Fig. 2 for example. So there are n objects with k partitions, dividing the data into groups, such that each group must have one object at minimum, Most of these types of methods are distance-based. Good partitioning is when objects in the same cluster are related or “close” to each other, after using relocating techniques for moving objects from one cluster to another, for example (K-means) [25].

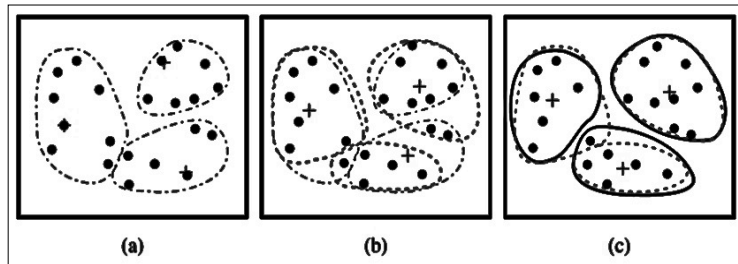


Figure 2 Partition based method example [25]

B- Hierarchical methods: This method generates an hierarchical decomposition for data objects in assumed data set as per Fig. 3 for example. This method is categorized into two types: agglomerative; divisive, based on the form of hierarchical decomposition [26]. The agglomerative approach, or “bottom-up approach” begins to form each object in separate groups and is successful when the objects merge or clusters come close to one another or a finishing state holds [25]. The divisive approach, or “top-down approach” begins with all the objects in the same group. Then with restatement, each object can be split from the cluster into a smaller one until all the objects fall into one cluster or a finishing state holds.

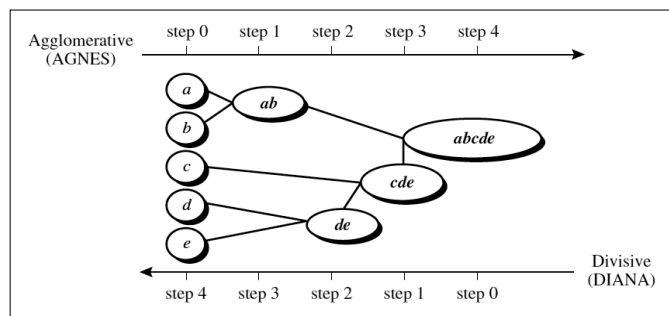


Figure 3 Hierarchical based method example [25]

C- Density-based methods: This method's idea is to continue aggregating clusters including the density (objects number) in the "neighborhood", to end up with sphere-shaped clusters but including some random-shapes. As shown in Fig. 4 for example. This method can separate a set of objects into many limited clusters, or a hierarchy of clusters. Normally, this method is good for limited clusters only [24].

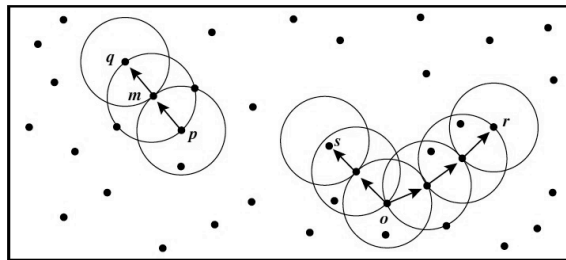


Figure 4 Density based method example [24]

D- Grid-based methods: This method makes the objects to appear as a finite number of cells formed as a grid structure. As shown in Fig. 5 for example. It is a fast processing approach, normally depending on the number of data objects and the number of cells. It is an effective method for many data mining problems and can be integrated with other clustering methods such as hierarchal or density [16].

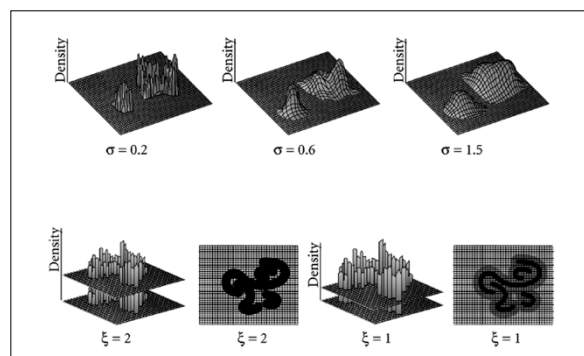


Figure 5 Grid based method example [16]

2.4 Cyber Attacks Types

This section presents many types of cyber attacks depend on attack targets, or network protocols attacks types, and malicious code types.

2.4.1 Attack types

A- Scanning Attack: this type of attack uses scanning techniques to understand what type of network the victim is using and then acquires information about that network such as network topology, operating system used, which server application is running, etc [7]. Then the attacker uses this information to launch attack by opening a TCP connection to send a SYN. If he receives an SYN acknowledgment this means that this port is listening and can be used to attack elsewhere if the port is not closed [27].

B- Denial of Service Attack: this type of attack has two types of DoS attacks (flooding, and flaw exploitations). Flooding: can be using with the ping command by sending crashing number of packets to the victim. Flaw exploitations: using the weak point on software to do the attacks and then exploit the system by freezing, rebooting or falling [28].

C- Penetration Attack: this type of attack includes attacks that give the unauthorized attacker access to the resources or to data. This type is able, using randomly execute codes, to work on the root easily, to give the attacker a way to affect the system and resources [4].

D- Spam: Many copies of the same message called Spam are flooding the Internet. Most of them are advertising of commercial things, the receiver pays rather than the sender, the technical features are related to both email and the web protocols (SMTP and HTTP) assume a low level of security [29]. Spam has two types of unusual effects on Internet users [30], “cancellable Usenet spam and Email spam”. Spam costs people using the Internet additional payments if they read it. In particular it charges for transmitting spam online services and ISPs, sending spam to mailing lists (public or private email). So they can attract mail lists of addresses, or using direct target to the mailing list as their method of attack.

2.4.2 Network protocols attacks

In this section, covering most network protocols can be target by attackers:

A- Internet Control Message Protocol (ICMP)

B- Transmission Control Protocol (TCP)

C- Address Resolution Protocol (ARP)

D- User Datagram Protocol (UDP)

E- Domain Name System (DNS)

A- ICMP: Is an information messages send to host which is used in IP layer, no authentication in this message so that leads attackers to used it, like denial-of-service attackers there are many kinds of attacks using the ICMP protocol [7]:

1- ICMP DOS Attack: attacker using “Time exceeded” or “Unreachable destination” with ICMP messages that makes a system in which the host fails in connection so the attacker sends faked messages to the host and gateway. Then he will redirect a message to both and can create a host to receive and send packets over that attacker’s host [27].

2- Ping of Death: when the attacker sends a request packet and the ICMP echo is greater than the size of the IP packets, this causes the receiver to fragment this packet. Then the target cannot reconstruct it and the OS will be rebooted or fail.

3- ICMP PING floods attack: in which a huge ping to a target's system make it unable to respond to messages to genuine traffic [30].

B- TCP: to start communication between two users application via TCP, the sender must send a request message to the receiver’s exact address, then do a handshake between the two users. Then the TCP sets up a full-duplex communication for users applications. This will remain open to users until one of them closes the application; TCP protocols have many security problems that are used by attackers [28]:

1- TCP SYN / TCP ACK Flood: A very public type of attack that tries to deny service. The attacks start as a typical TCP connection then the network server with the client start to exchange the data and information using TCP packets. The client computer will continue sending ACK to the server, the ACK insures the server connection is continually demanded [7]. The server replies with a ACK to the client.

The client uses this method to send other packets to be accepted. This begins the session, and then the client will send and receive without any need to open a new session. This causes the server to have an open session while waiting for the final packet- this makes the server to fill up the existing connection and deny any other client demand processing.

2- TCP Sequence Number Attack: in this type of attack to get control of the session, the attacker interrupts and then rejoins a number similar to the original sequence number. This can hijack the connection sessions if the valid sequence number is in the right sequence between server-client sessions so it can seize the data about the network [29].

3- TCP Hijacking or active sniffing: it involves the attacker gaining access to the client in the network and make it disconnect properly from the network It then inserts another machine with the same IP address as the attacker to the network. This step, if done very quickly, gives the attacker control and access to the session and to get all the information about the original network and system [30].

4- TCP reset attack: or “forged TCP resets” or “spoofed TCP reset packets” or "TCP reset attacks". These terms denote a technique of influencing using Internet communications.

C- ARP: to find any address in a network by using the name of the address you need to map it by this protocol which matches the IP addresses [7] to the data link address:

1- ARP flooding: ARP packet uses system resources so it cannot take more memory size. Then a large number of these packets cannot be processed efficiently. An attacker sends a large number of this type of packets with different IP addresses to flood the target. The victim cannot find the address so it makes a disconnect. In addition attacker can send a large number of such packets with non-resolved IP addresses. This makes process work in an open loop without solving the destination IP address and halts the system.

2- User spoofing: a forged ARP packet sent by an attacker with a wrong IP to an MAC address. The forged ARP is sent from the host to cheat the router gateway adding this mistake to the host, then the normal communication between gateway and victim host are interrupted [31].

3- Connection Hijacking and Interception Packet interruption: in which a host can be used by an attacker to manipulate a possible way to get connection control.

4- In DoS attack: victims are prevented from communicating with the Internet, or with each other. This is done by corrupting their ARP caches with forged items involving nonexistent MAC addresses, or in the malicious host disabling the IP packet routing option, so any redirected traffic going in a wrong way [28].

D- UDP: for easy transmission without using handshaking for media files or video, so these transmissions do not need to be in order, reliable, and data integrity. So the data can be received out of order, or with missing packets or duplications. UDP assumes that correction and error checking is not necessary to avoid overload in the applications for some processing in the network interface, yielding attacks like [7]:

1- UDP flood attack: Similar to ICMP flood attack, this attack sends a large number of UDP messages packets to the target in a short time, so that makes the target too busy and unable transmit the normal data packets to the network [27].

2- Fraggle: similar to a smurfing attack, with the difference that it uses the User Datagram Protocol (UDP) instead of ICMP.

3- Teardrop: (A teardrop type of DoS attack), this attack works by degrading the offset data in UDP packet: this makes it difficult to restore it into the original packets [31].

E- DNS: Security and Pharming

1- Phishing attack: the user will be deceived into visiting a fake page by using scam email. If user starts to notice the URL, they will find the URL is not the original for that site. So he knows to recognize and detect the phishing attack by URL. But a hacker can take the phishing attack to another level with the pharming attack [31].

2- Pharming attack: is to redirect to the fake (phishing) page - nevertheless the host enters the true address. The pharming duration is a consequence of pharming and phishing. Pharming and phishing have been used for online character theft information in recent years [30]. Pharming has become a major threat to businesses and online finance websites. There are a number of other variants on this theme, including feeding false DNS records to genuine servers. Older DNS servers would

accept additional records without checking; if they asked your server where X was, you could volunteer an IP address for Y as well [29].

2.4.3 Malicious code types

Programming code that causes damage to a computer or system is called malicious code. This code can't work easily or unassisted if the system has anti-virus tools [29]. Malicious code can activate itself or need an action to be performed to work, such as clicking on or opening an attachment in email [30].

Malicious code can do more than affect one computer. It can also slip into networks and spread. It can steal information or send messages through email, and delete files [31]. Malicious code can be found in “the form of scripting languages, ActiveX controls, browser plug-ins, Java applets and more” [29].

Malicious code can find in different forms:

- **Virus:** is a public type of malicious code, it is a small piece of programming code attached to other authorized files or genuine programs. When it runs it will copy itself onto the system in the user computer and can be extended to other computers in the same network. Viruses can range from being quite harmless to causing extensive damage to a system [28].
- **Worms:** are pieces of malicious code making copies of itself. Conditions have to be right for a worm to proliferate. It exploits a sum of vulnerabilities to propagate from one computer to other in the system network. It may leave a copy in the memory to run- some worms do not harm the victim computer system or files [30].
- **Trojan:** these horses are a type of malicious code resembling harmless software. This is a ruse to get into a computer. The Trojan hides inside a safe program and can be installed with that program. Sometimes they offer somebody in control of the victim's computer a remote location [27].

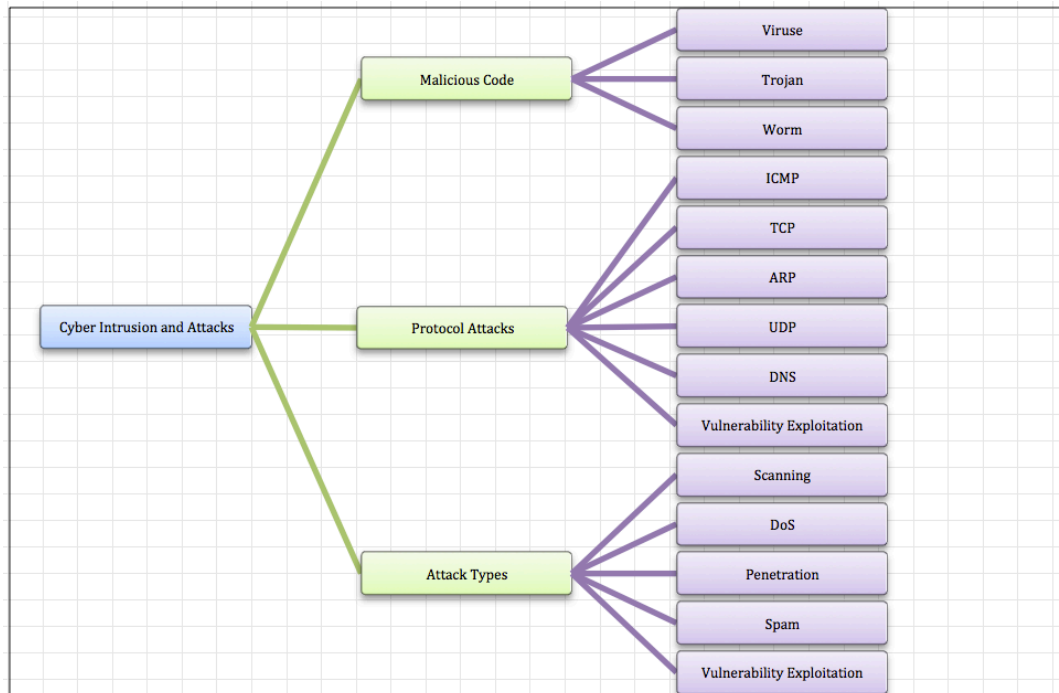


Figure 6 Cyber intrusion and attacks

2.5 Intrusion Detection Systems

IDS: this system is used for data analyzes to detect intrusions or malicious activity, and then send detection alarm report when activity is found. This section presents how IDS work and discusses many types of IDS classifications.

2.5.1 Characteristic approaches

IDSs typically contain several components. Classical IDS architecture consists of four components [32], shown in Fig. 7 decoder, preprocessor, detection engine and alert module [33].

IDS components description:

- The decoder: to transform audit data into data that can be accessed by the preprocessor. It obtains pieces of raw audit data from data collectors [33].

- The preprocessor: to obtain features from data, the TCP preprocessor is the most commonly used in network intrusion detection that can compile session flow for TCP segment set in that data [34].
- The detection engine: to examine the intrusion by searching in the data that prepared by preprocessor, if it finds an intrusion it will request an alert module [35].

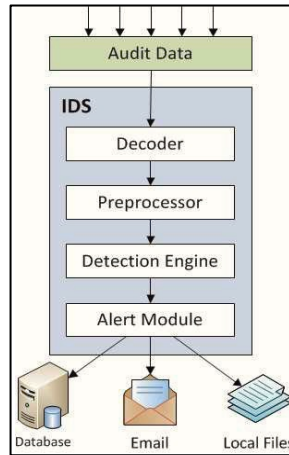


Figure 7 IDS component [32]

- The alert module: to raise the alerts requested by the detection engine. Raising an alert can be the choice of the system administrator emailing [36].

2.5.2 IDS classifications approaches

IDSs has many types of classification depending and matching an IDS component:

1-Concerning the source of data, (Network-based and Host-based):

- Network IDSs (NIDSs): to analyze network traffic, they work by analyzing network traffic in high OSI model layers like transport and application with different detection levels, HTTP is a preprocessing protocol with NIDS. NIDSs are typically located outside of the system therefore enabled to monitor a complete Local Area Network (LAN) [37].
- Host IDSs (HIDSs): to analyze data limited to the devices. By analyzing system calls the sequence of running programs in the device [35]. An ideal HIDS analyzes many things related to the device running programs like “system call arguments, memory registers, stack states, system logs, user behaviors, etc” [33].

2-Concerning to the detection model, (Misuse-based, Anomaly-based and Hybrid):
Will explain with more details in detection part.

3-Concerning to the type of action, (Active and Passive):

- Passive IDS: when it detects malicious behavior and an alert is raised.
- Active IDS: to execute a predefined action it tries to counterbalance the malicious data. Many authors denote an active IDS as Intrusion Prevention System (IPS).

Furthermore, IDS has many other additional classifications. Firstly, the technology may be “Wired and Wireless” [37]. Also, Wireless IDS can be more classified as “Fixed or Mobile”. Secondly, concerning the data processing method, it can be classified as “Centralized, Distributed and Hierarchical”. Concerning the detection process, timing can be classified as “Real time and Non-real time” [37]. Finally, concerning the detection technique, “State-based and Transition-based”.

2.5.3 Detection approaches

There are many approaches offered. They can be classified into three main categories: misuse, anomaly and hybrid detection. Each of these detection approaches can work together or can be integrated with the machine learning technique used for anomaly detection [37].

A- Misuse detection: or rule detection monitoring the events by earlier knowledge from known malicious activity and attacks. It compares the monitored events with intrusive patterns stored in the detection database [35]. These patterns are called signatures, and misuse detection can also be called signature-based detection. Snort is signature based [33] and contains a huge number of freely available signatures. The signatures can be in many different formats, and accepted for a deep inspection of different protocols in network layers (IP): moreover, in a transport layer TCP and UDP next in a high layer-like application layer such as HTTP, FTP, SMTP, etc [33].

For misuse detection, the signature-based method is the best common approach, but another method to represent knowledge is the attack path analysis [37]. Using many attack paths for modeling actions to detect intrusions, if it is monitored carefully it

can follow the attacker's path. Misuse detection is not able to detect zero-day attacks because these attacks do not have a related signature in the IDS, or IDS is not updated to find new attacks that demand new signatures.

B- Anomaly detector or behavior detection: comparing monitored action with a normal model for detects intrusions. It computes the normal model by learning process that is typically done off-line. Monitoring can be for “network flows, service requests, packet headers, data payloads, etc”. During the learning process, to compute the normal model the system will analyze a set of normal data [38].

To compute the model from network traffic data there are different approaches:

- Statistic-based approaches: the normal model can be defined as the probabilities of appearance of certain patterns in the training data [34], covering simple statistical and thresholds operators such as the standard deviation, mean, co-variance, etc. At the time of detection, each activity that noticeably differs from the learned probabilities is measured as malicious.
- Specification-based approaches: specialists who identify how the system can be monitored and how it behaves: they build a range of methods [39]. Such methods are considered anomalous. Such methods can use the state-machine specifications from network protocols.
- Heuristic-based approaches: by using machine learning algorithms generate the model of normal behavior automatically [39], or by using evolutionary systems [36] or using artificial intelligence methods [34]. This is possibly the most used approach in the research group.
- Payload-based approaches: this method analyzes data from application layer like HTTP protocol to detect and find attacks by using n-grams for anomaly detection for detecting malicious payloads to deriving features from the monitored data [34]. A series of n sequential bytes is derived from a long string called an n-gram.

C- Hybrid Detection: Collecting data has many false positives and this leads to some problems with anomaly-based detectors [33]. So use misuse based detectors can solve this issue but still cannot detect a zero Day attack. Using anomaly-based detection can prevent these types of attacks [35]. A combination of both techniques is necessary. Hybrid IDSs combines both misuse detection and anomaly detection [39]. The data preprocessor does anomaly-based detection while detection engine does the signature matching [38]. Finally, Fig. 8 summarizes the IDS in types and categories.

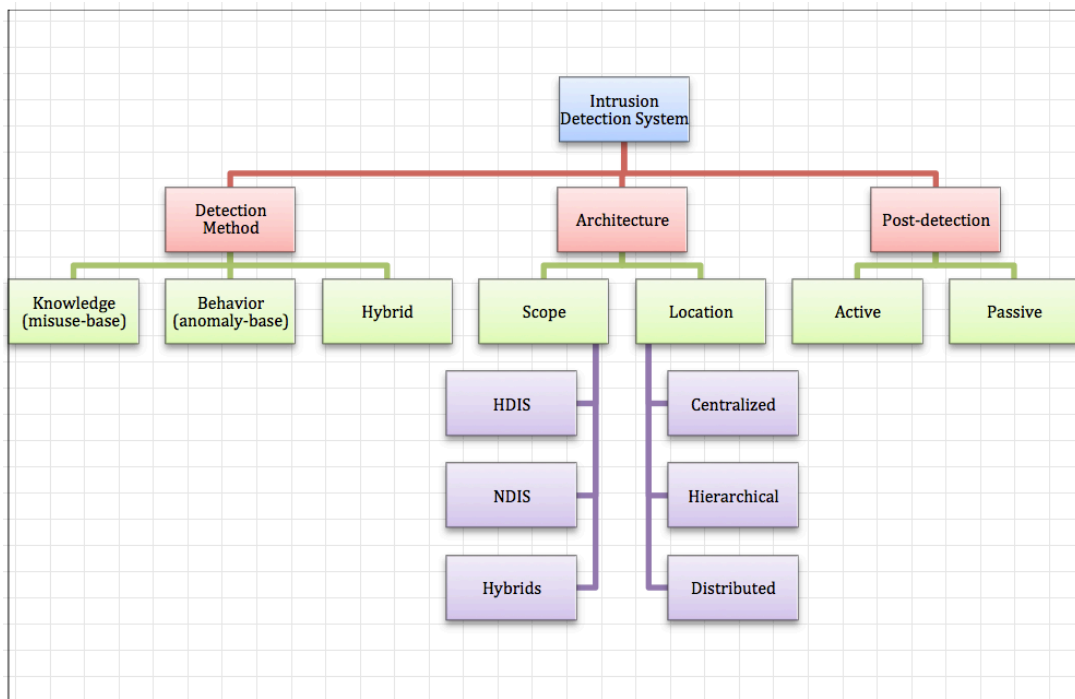


Figure 8 IDS categorization

CHAPTER 3

DATASET AND METHOD

This chapter presents materials and methods. Firstly, Section 3.1 presents DARPA'99 Dataset. Next, Section 3.2 presents a DBSCAN algorithm. Then, Section 3.3 presents DenStream algorithm. Finally, Section 3.4 presents sensitivity and specificity.

3.1 DARPA Dataset

DARPA'99 dataset is used as a data depository from the Defense Advanced Research Projects Agency (DARPA) in this link:

<http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1999data.html> as a sample of normal network traffic [40]. The simulation of network communication collects this dataset from a fictitious United States air force base. Specifically, this dataset is organized in two types: the first is an attack-free dataset without any anomalous, the second is attack-contained that is anomalous [40]. Thus, for this thesis to work one must select the first week Monday for a training dataset without attack and select the fourth week Friday for testing a dataset containing attack packets of DARPA'99 outside tcpdump.

Attacks in this dataset categorized into many kinds that can find on this link: http://www.ll.mit.edu/ideval/files/master_identifications.list, [41]. As selected the dataset also selects attacks from the attack list tables for that day of testing. The dataset DARPA'99 of this thesis for anomalous packet detection has many types of attacks as R2L (Remote to Local), U2R (User to Root), DoS (Denial of Service) and Probe [42].

- Remote to Local (R2L): attacker works to gain access to a system he doesn't have a key for.

- User to Root (U2R): attacker has a limited key to use the machine and is working to gain a super user account.
- Denial of Service (DoS): attacker works to block the authorized user from using a service.
- Probe: attacker works to get more information about target machine [43].

In this study, as the main idea analyzing the stream of data for network traffic using n-gram method for extracting features from the dataset file used “the feature vector is the relative frequency count of each n-gram that is calculated by dividing the number of occurrences of each n-gram by the total number of n-grams” [44]. This study uses the 1-gram to calculate the average frequency of each character as ASCII from 0-255 [6], feature vector $x_p = (x_1, x_2, \dots, x_{256})$ where x_i contains the number of amounts of the i^{th} 1-gram in the method, path, and arguments of p [45]. Furthermore, it presents two procedures based on Density based clustering algorithms, firstly using DBSCAN and second using DenStream.

3.2 DBSCAN Algorithm

Definition: Density-Based Spatial Clustering and Application with Noise (DBSCAN): is an algorithm for clustering based on density. In addition one of the most popular approaches of data mining is clustering in this study presented density-based approaches relate a local cluster condition [45]. Dividing the data space into regions without any overlapping in one region, which objects similar to each other in one cluster can be separated to low object density regions (low object density called noise) [46]. A typical way to create regions with high density is to find an area with cell densities so the cluster is growing in a high-density area. In DBSCAN algorithm there are two important objects: clusters and noise. It can locate all clusters suitably, which autonomous of the shape, size, and location for everyone in clusters. DBSCAN is built on two typical concepts: density spread ability and density related ability [47]. Both of these concepts are linked to two input parameters: (ϵ) is the epsilon and (mp) is the clustering minimum points, also it has a core point with numerous points inside epsilon. The neighborhood shapes are determined by using the distance function for two points p and q by using Eq. (3.1).

$$N_s(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}, \quad (3.1)$$

Description: in this thesis as the first Phase, uses dbscan for clustering network traffic to detection anomalous or intrusions this algorithm creates the normal model. It then will use test dataset to examine this model input file for the algorithm contains normal and abnormal packets. The output file is predictions of normal and abnormal packets. Finally, it finds the best output results that will be discussed in the next chapter.

Algorithm 1 Using DBSCAN

DBSCAN algorithm	
1	DecideDBScanParams(epsmin, epsmax, epsstep, minPtsmax,
2	minPtsmin, minPtsstep, Ntrain, Ntest, max_i, max_j)
3	begin
4	maxfmeasure=-1;
5	for i=minPtsmin to minPtsmax with minPtsstep
6	for j=epsmin to epsmax with epsstep
7	begin
8	fmeasure1=DBScan(i,j, Ntrain)
9	fmeasure2=DBScan(i,j, Ntest)
10	if fmeasure2> maxfmeasure then
11	begin
12	maxfmeasure=fmeasure2
13	max_i=i max_j=j
14	end
15	end
16	return [max_i, max_j]
17	end

3.3 DenStream Algorithm

DenStream algorithm: is a density base algorithm, using the damped window model [48] to direct the idea that current data objects further than older ones. The weight is a measure of data objects that reduce over time exponentially. It expands as a function that used in DBSCAN for ε - neighborhood under the name of fading function. It can make updates to the information of data stream by Eq. (3.2)

$$f(t) = 2^{-\lambda t}, \quad (3.2)$$

where λ referred to the decay factor and t referred to the time, and can create a “core-micro-cluster” [49] by using three other attributes additionally for weight, center, and radius as more details. All these can be formally according to the step of time as for weight w

$$w = \sum_{i=1}^n f(t - T_i), \quad (3.3)$$

for center c

$$c = \frac{\sum_{i=1}^n f(t - T_i)p_i}{w}, \quad (3.4)$$

and for the radius r

$$r = \frac{\sum_{i=1}^n f(t - T_i)\text{dist}(p_i, c)}{w}, \quad (3.5)$$

The algorithm to work need to initialized with the DBSCAN algorithm [5] functional to the first InitN each points of data, which produces initial p -micro-clusters. After the initialization period for each new point of data, a merging algorithm is used to maintain the micro-clusters [49]. When a clustering request is received, the offline part of the algorithm generates the final clusters using a variant of the DBSCAN algorithm.

Description: in this thesis as the second Phase, it uses DenStream for clustering network traffic packets to detect the anomalous packets by creating a normal model and then examines the test dataset with this model. The input file for the algorithm contains normal and abnormal packets. The output file is predictions of normal and abnormal packets. Finally, getting the best result for this detection and more details will be discussed in the next chapter.

Algorithm 2 Using DenStream

	DenStream algorithm
1	DecideDenStreamParams(epsmin,epsmax,epsstep, minPtsmin,
2	minPtsmax, minPtsstep, denepsmin, denepsmax, denepsstep, dfactor,
3	initPntrate, Ntrain, Ntest, max_i, max_j)
4	begin dbscan initialize
5	maxfmeasure=-1;
6	for i=minPtsmin to minPtsmax with minPtsstep
7	for j=epsmin to epsmax with epsstep
8	begin
9	fmeasure1=DBScan(i,j, Ntrain)
10	fmeasure2=DBScan(i,j, Ntest)
11	if fmeasure2> maxfmeasure then
12	begin
13	maxfmeasure=fmeasure2
14	max_i=i max_j=j
15	end
16	end
17	return [max_i, max_j]
18	end
19	begin Denstream
20	for k=denepsmin to deneps max with deneosstep
21	fmeasure1=DenStream(k,Ntrain)
22	fmeasure2=DenStream(k,Ntest)
23	if fmeasure2> maxfmeasure then
24	begin
25	maxfmeasure=fmeasure2
26	end
27	dfactor = 2.0
28	initPntrate = 0.4
29	return [max_i, max_j]

3.4 Sensitivity and Specificity

As a mean point in binary classification calculating the result or how to measure the quality of the work or the test in this issue the sensitivity is a statistical measure of the implementation, also called “classification function” [50] sensitivity can be measured by computing the true positive rate of the test, it is the measurement of the amount of positive correctly recognized or real positive. Specificity can be measured by computing the true negative rate of the test it is the measurement of the amount of negatives correctly recognized, and the four outcomes are shown in Fig. 9.

		Condition (as determined by "Gold standard")			
		Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{Condition positive}}{\Sigma \text{Total population}}$	
Test outcome	Test outcome positive	True positive	False positive (Type I error)	Positive predictive value (PPV, Precision) = $\frac{\Sigma \text{True positive}}{\Sigma \text{Test outcome positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{False positive}}{\Sigma \text{Test outcome positive}}$
	Test outcome negative	False negative (Type II error)	True negative	False omission rate (FOR) = $\frac{\Sigma \text{False negative}}{\Sigma \text{Test outcome negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{True negative}}{\Sigma \text{Test outcome negative}}$
	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	True positive rate (TPR, Sensitivity, Recall) = $\frac{\Sigma \text{True positive}}{\Sigma \text{Condition positive}}$	False positive rate (FPR, Fall-out) = $\frac{\Sigma \text{False positive}}{\Sigma \text{Condition negative}}$	Accuracy (ACC) = $\frac{\Sigma \text{True positive} + \Sigma \text{True negative}}{\Sigma \text{Total population}}$	
	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	False negative rate (FNR) = $\frac{\Sigma \text{False negative}}{\Sigma \text{Condition positive}}$	True negative rate (TNR, Specificity, SPC) = $\frac{\Sigma \text{True negative}}{\Sigma \text{Condition negative}}$		
	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$				

Figure 9 2X2 Conditions and test outcome [52]

As a note, the rate of perfect sensitivity is described as 100% sensitive and 100% specific theoretically [51], but in practical fields and issue can be the sensitive as a maximum value of test sensitivity. In addition there are many other equations for calculating dependent on four main values of the test. These four outcomes can be shown in Fig. 9 to analyze the output results.

Typically the correctly recognized is True Positive (TP), incorrectly recognized is False Positive (FP), correctly refused is True Negative (TN), and the incorrectly refused is False Negative (FN). Importantly, in the study used Positive as Attack and the Negative as Normal [52].

The test's accuracy is measured by a statistical analysis of binary classification formula known as F1 score or (F-measure). The F-measure can be understood as an average weighted of the sensitivity and precision, where a best value of F-measure reaches to 1 and the worst value reaches to 0. To get all the results in this work we used the formula and equations denoted in the Fig. 9 as Eq. from (3.6) to (3.11) [52]. For computing the output result for detecting the anomalous of the network packets by applying two density algorithms used for sensitivity used Eq. (3.6)

$$\text{Sensitivity or TPR} = \frac{TP}{TP + FN} \quad (3.6)$$

For specificity used Eq. (3.7)

$$\text{Specificity or SPC} = \frac{TN}{FP + TN} \quad (3.7)$$

For precision used Eq. (3.8)

$$\text{Precision or PPV} = \frac{TP}{TP + FP} \quad (3.8)$$

For False Positive Rate used Eq. (3.9)

$$\text{FPR} = \frac{FP}{FP + TN} \quad (3.9)$$

For accuracy used Eq. (3.10)

$$\text{Accuracy or ACC} = \frac{(TP + TN)}{(TP + FN)(FP + TN)} \quad (3.10)$$

And for the F1 score that is the main formula that used to select the best output result from the output of the testing the algorithms we used Eq. (3.11)

$$\text{F1 score or F_measure} = \frac{2TP}{(2TP + FP + FN)} \quad (3.11)$$

CHAPTER 4

EXPERIMENTS AND RESULTS

This chapter presents experimentation results of this study in three sections. Firstly, Section 4.1 presents experiments of DBSCAN algorithm. Next, Section 4.2 presents experiments of DenStream algorithm. Finally, Section 4.3 Experiments over 2000 packets.

4.1 Experiments of DBSCAN Algorithm

This section presents the results of detecting anomaly network packets, using data mining techniques in two stages. We have used DARPA dataset. The first stage uses DBSCAN algorithm experiments for clustering normal data and classifying mixed data type consisting of normal and attack data. Next, we analyze the results.

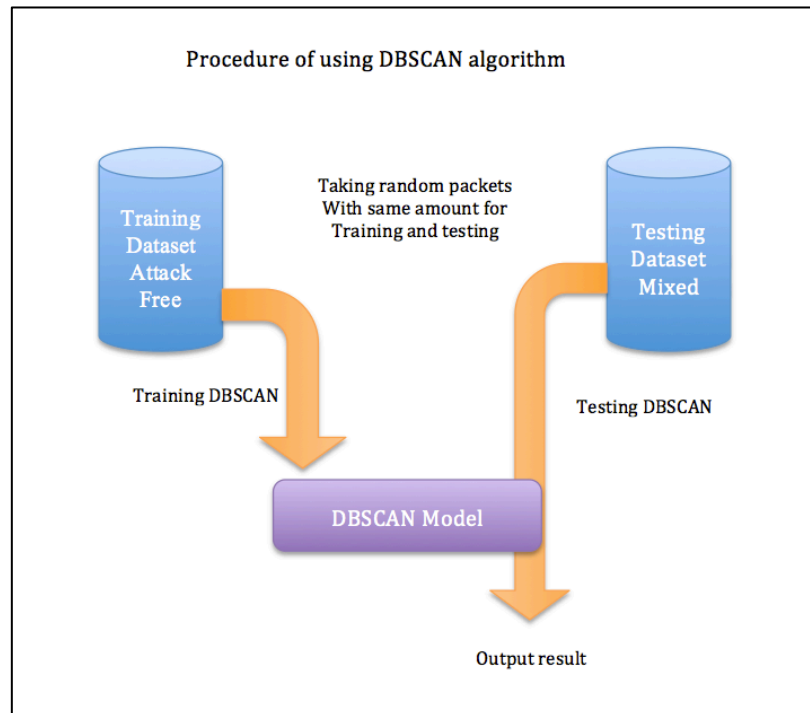


Figure 10 DBSCAN procedure

Phase 1: First, a typical model from training dataset having attack-free type by taking random 1000 packets from week1/Monday. Assuming the two input parameters of DBSCAN (e) and (mp). The range value of (e) is from 1 to 10 with 0.5 steps, and the range value of (mp) is from 1 to 10 with 1 step. Output result by running the algorithm in a multi-run mode for testing random 1000 packets from week4/Friday attack contained. For testing dataset for detecting the anomaly network packets this step done by 38680 s and the result of F1 as shown in Fig. 11.

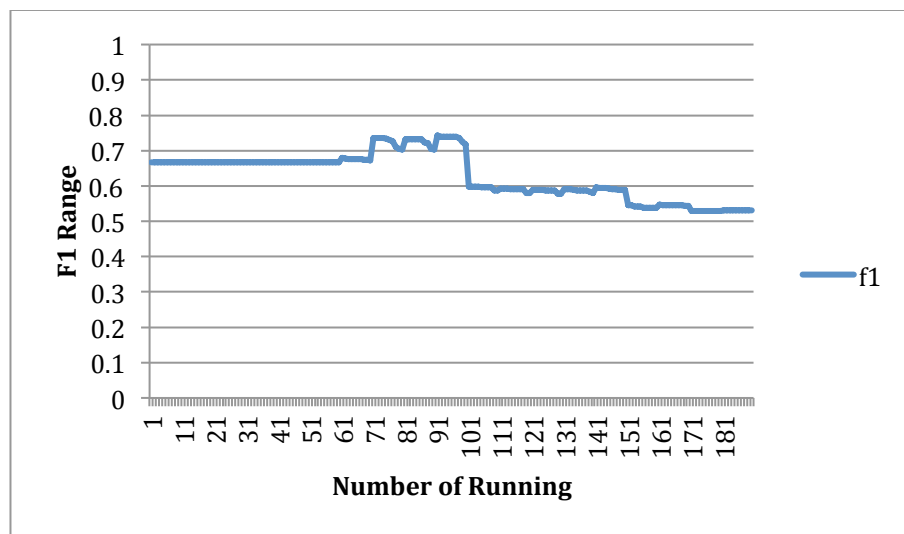


Figure 11 F1 score for DBSCAN

Next, output result of F1 has been analyzed. The highest output result of F1 has been given in Table 1.

Table 1 Highest Output Result of F1

Epsilon	MinPoints	F1
4.5	3	0.735742972
4.5	4	0.735742972
4.5	5	0.733974359
5.5	1	0.742349457
5.5	2	0.74015748
5.5	5	0.739429695
5.5	6	0.739429695

Phase 2: The first step after getting the output result Table 2, three cases has been randomly created with respect to highest values of F1. Then (e) and (mp) values are investigated by running the algorithm. For each attack categories (R2L, U2R, DoS and probe), these cases have been investigated. The output results have been analyzed.

These three cases of the parameters for varying (e) and (mp) have been given in Table 2.

Table 2 Cases of DBSCAN

	R2L		U2R		DoS		Probe	
	E	MP	E	MP	E	MP	E	MP
Case1	4.5	8	4.5	3	4.5	4	4.5	2
Case2	5	3	4.5	10	5	1	5	4
Case3	5.5	6	4.5	6	4.5	8	5.5	5

A mixed data consisting of attack type is classified and F1 results have been given in Table 3.

Table 3 Output Results of F1 with Attacks

	R2L	U2R	DoS	Probe
Case1	0.766	0.673	0.610	0.8071
Case2	0.787	0.650	0.600	0.790
Case3	0.739	0.664	0.609	0.795

By referring to Table 3 of F1 score values. It can evaluate the approximate value 0.8 for a probe, approximate value for R2L is 0.75, approximate value for U2R is 0.65 and the approximate value for DoS is 0.6. It is the minimum F1 value obtained for classification among different attack types. The output result for F1 has been shown in Fig. 12.

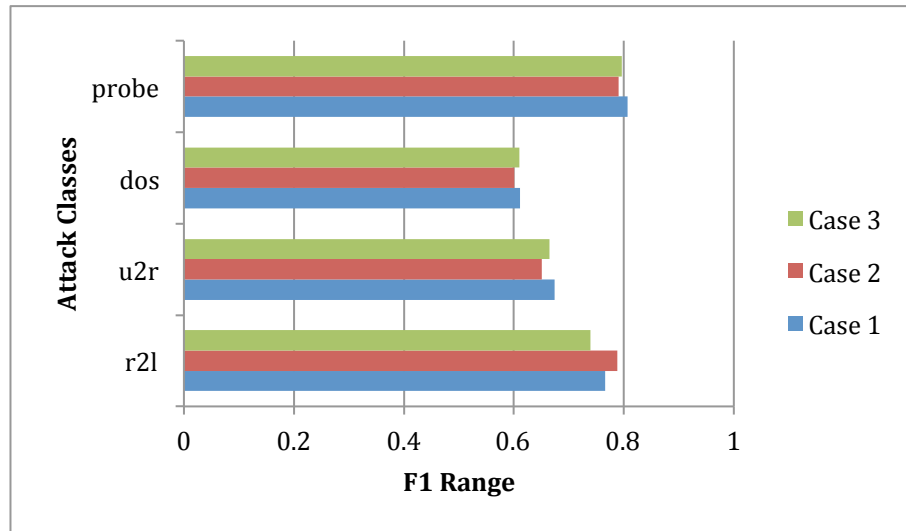


Figure 12 Output result for F1 with attacks

TPR results have been given in Table 4. 1000 random packets for training and 1000 random packets have been used for testing and obtaining the sensitivity results for DBSCAN algorithm.

Table 4 Output Result of TPR

	R2L	U2R	DoS	Probe
Case1	0.812	0.788	0.614	0.809
Case2	0.780	0.794	0.578	0.803
Case3	0.752	0.788	0.614	0.807

When the sensitivity of DBSCAN algorithm for attack detection is analyzed, the approximate value for detecting a probe is 0.80. The approximate value for R2L is 0.78 and the approximate value for detecting U2R is 0.78, but the sensitivity for the DoS is as the least among all, which is approximately 0.60. The sensitivity output result has also been shown in Fig. 13.

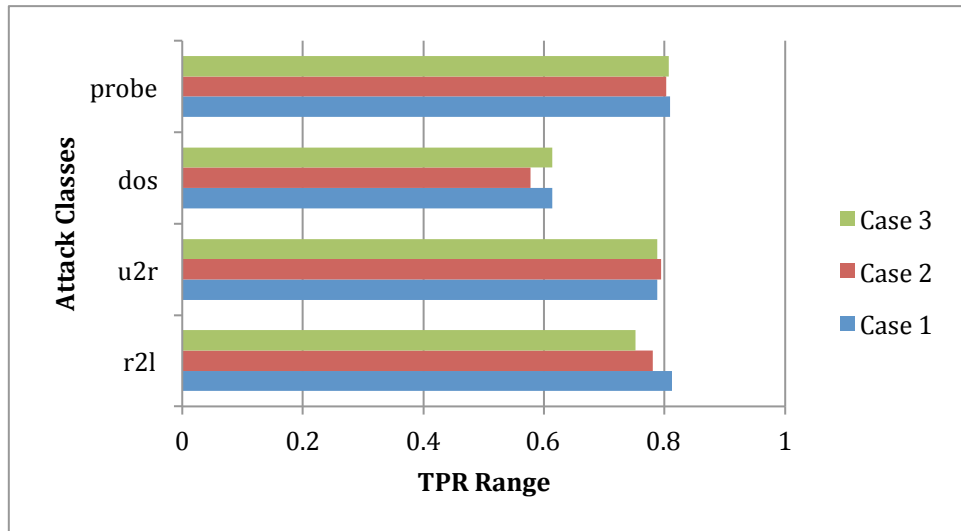


Figure 13 Sensitivity of DBSCAN

TP results have also been studied and given in Fig. 14. DBSCAN algorithm has been used for classification and according to the results; DoS type attack obtained the minimum results where as higher scores have been obtained for probe and R2L.

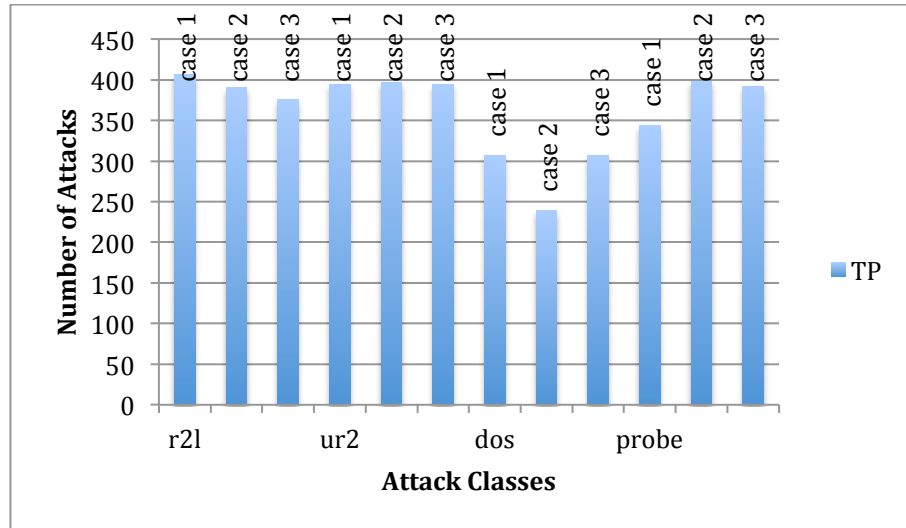


Figure 14 Attack detected with DBSCAN

Table 5 summarizes the output result values for TPR and FPR, after running DBSCAN algorithm in single run mode, runtime duration for the classification is around 95s for all different attack types.

Table 5 TPR and FPR of DBSCAN

Attack	TPR	FPR
R2L	0.812	0.174
	0.780	0.204
	0.752	0.282
U2R	0.788	0.174
	0.794	0.204
	0.788	0.282
DoS	0.614	0.174
	0.578	0.204
	0.614	0.282
Probe	0.809	0.174
	0.803	0.204
	0.807	0.282

Fig. 15 illustrates the TPR and FPR testing results for all cases. As shown in Fig 15 has some restriction with DoS attack type.

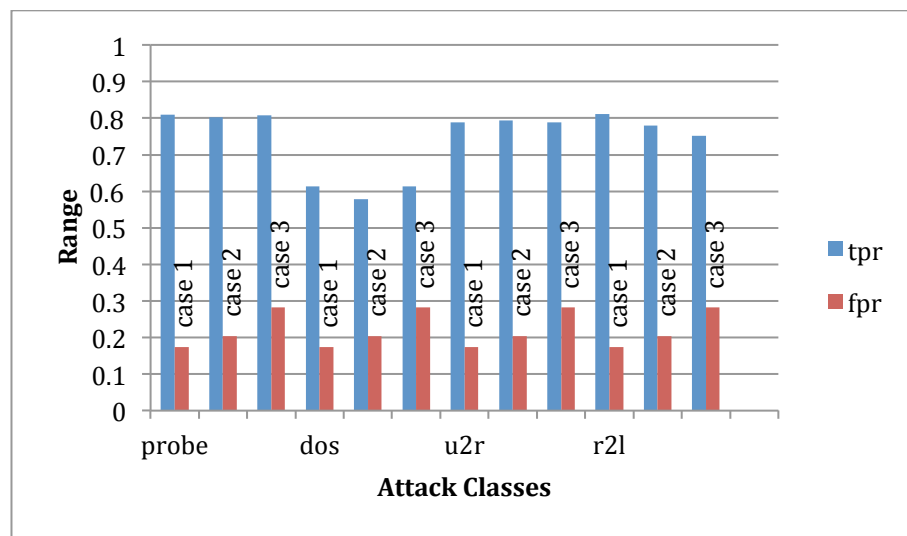


Figure 15 TPR and FPR of DBSCAN

After studying all the results for the three cases by using DBSCAN algorithm, Data having DoS type attacks got less sensitivity and F1 scores when compared to other attack types. Meanwhile, DBSCAN has given better results in terms of sensitivity and F1 for detecting R2L and probe attacks individually.

DBSCAN algorithm in single running mode has been presented in 4.1. DBSCAN algorithm costs more since testing with fine-tuned parameters is our goal and scans the possible value sets are scanned for this purpose. The elapsed times for detecting anomalous with both algorithms have been measured in terms of second. Elapsed times for all attack types in different cases have been summarized in Table 6.

Table 6 Duration Process Time

	R2L	U2R	DoS	Probe
Case1	81s	83s	97s	83s
Case2	88s	80s	101s	85s
Case3	98s	82s	95s	99s

4.2 Experiments of DenStream Algorithm

In this part of the study, another algorithm called DenStream has been studied to detect anomalous network packets using the DenStream algorithm. This system can either run on a particular parameter configuration or scan a parameter search space. An exhaustive search on parameter space has been performed.

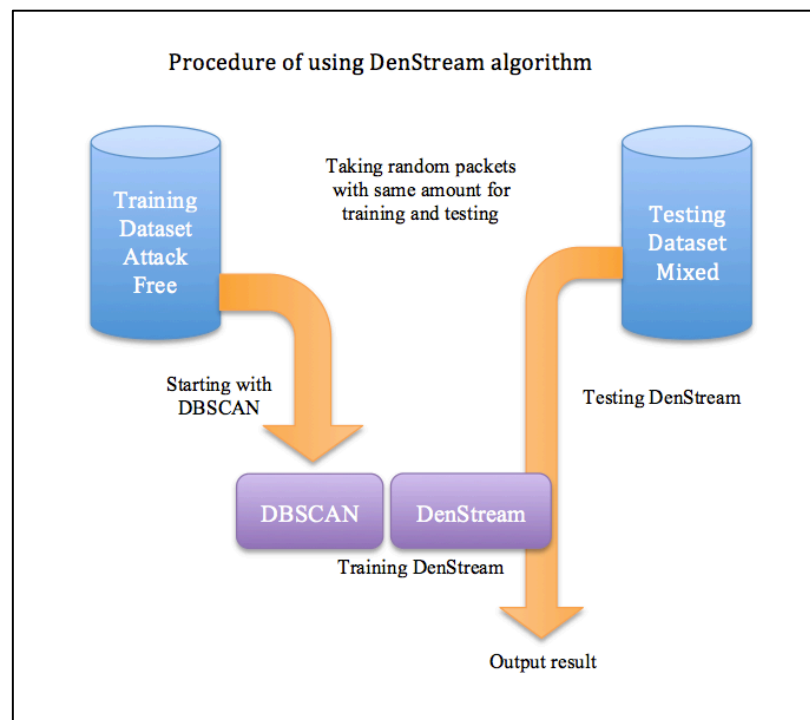


Figure 16 DenStream procedure

First, a normal model by using attack-free dataset has been as the training data. A model is built. Taking 1000 packets randomly from week1/Monday has created attack-free dataset. In addition, in the DenStream algorithm, one must initialize the model with DBSCAN. Training dataset has been divided into two parts $t1$ as 40% of the dataset and $t2$ as 60%. $t1$ part has been used for performing DBSCAN algorithm. Later, the process continues with clustering $t2$ with DenStream in order to create a normal model. In the training model, there are five parameters belonging to both algorithms, two of them are the same as DBSCAN and the others are for DenStream algorithm. d_factor value has been assigned to 2.0; $init_point_rate$ has been assigned to 0.4. DenStream epsilon range value from 2 to 11 with 0.5 step value has been studied; we have scanned the parameter search space by running with varying parameter values for DenStream algorithm. 1000 packets have been selected randomly from dataset week4/Friday. It contained the attack type data to classify normal and attack type (anomalous) network packets, the output results for F1 have been shown in Fig. 17.

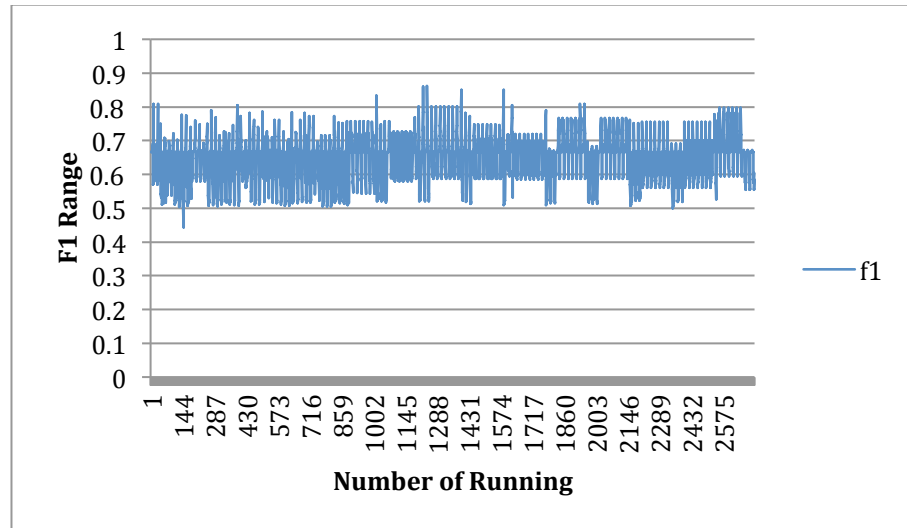


Figure 17 F1 output in DenStream

After analyzing the output results, the highest values according to F1 have been reported in Table 7.

Table 7 Highest Output Result of F1

Epsilon	DB.minpoints	DB.epsilon	F1
8.5	1	2	0.809484873
8.5	1	2.5	0.809484873
6.5	8	4	0.805369128
6.5	1	6	0.834042553
6.5	1	6.5	0.800788955
6.5	2	6.5	0.860335196
6.5	3	6.5	0.861136999
7	4	6.5	0.800331400
7	5	6.5	0.800331400
7	6	6.5	0.800331400
7	7	6.5	0.800331400
7	8	6.5	0.800331400
7	9	6.5	0.800331400
7	10	6.5	0.800331400
7	1	7	0.851449275
7	1	7.5	0.850909091
7.5	3	7.5	0.804995197

Table 8 reports results for three random cases having the highest values of F1 table by examining (ϵ) DenStream parameters after running the algorithm. This step has been repeated three times for each attack categories (R2L, U2R, DoS and probe).

Table 8 Cases of DenStream

	R2L			U2R			DoS			Probe		
	E D	E DB	MP	E D	E DB	MP	E D	E DB	MP	E D	E DB	MP
Case 1	6.5	6	1	5	4.5	3	6.5	6.5	3	6.5	1	1
Case 2	7.5	7.5	3	5.5	5.5	1	6.5	5	5	7	3	3
Case 3	8.5	8	9	5.5	3.5	4	5.5	5	5	7.5	1	1

Next, F1 and TPR scores have been analyzed. They have been given under tables according to three cases of (ϵ) of DenStream and (ϵ , mp) of DBSCAN for different attack types; F1 scores can be followed in Table 9. These scores consist of the aforementioned cases and attack types. Results have also been illustrated in Fig. 18.

Table 9 Output Result of F1

	R2L	U2R	DoS	Probe
Case 1	0.834	0.765	0.621	0.885
Case 2	0.840	0.763	0.666	0.883
Case 3	0.809	0.771	0.655	0.801

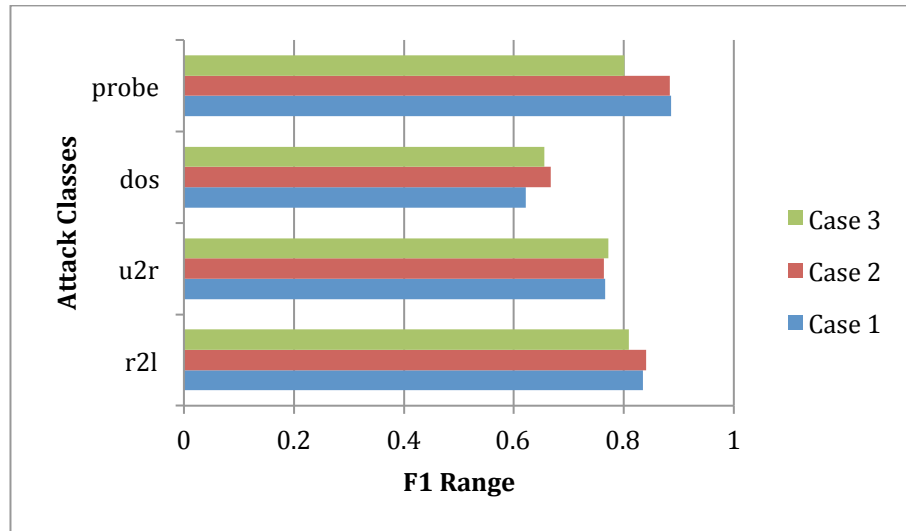


Figure 18 Output result for F1 with attacks

with reference to Table 9 and Fig. 18, F1 score for probe data is 0.87, R2L is 0.82 and U2R is 0.75 and DoS is 0.65. DoS is the minimum among all.

TPR results have been given in Table 10. This concerns the sensitivity of the running DenStream algorithm on 1000 random packets for training and 1000 random packets for testing.

Table 10 Output Result of TPR

	R2L	U2R	DoS	Probe
Case 1	0.980	0.892	0.656	0.963
Case 2	0.946	0.886	0.772	0.946
Case 3	0.990	0.894	0.660	0.976

The detection sensitivity of DenStream algorithm is 0.95 for probe and R2L. U2R has the value 0.90. But the sensitivity for the DoS is less than all others, which is 0.70. The sensitivity result has been shown in Fig. 19.

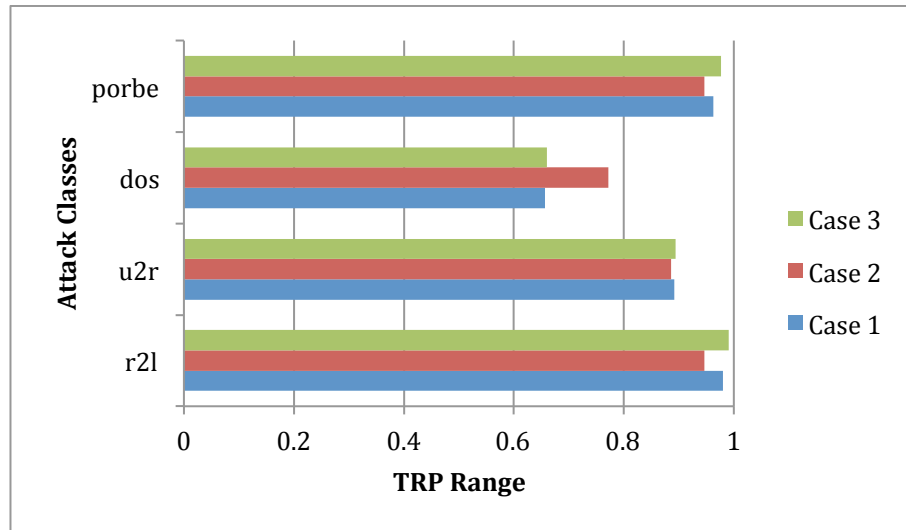


Figure 19 Sensitivity of DenStream

Fig. 20 shows the numbers for attacks detected (TP) by using the DenStream algorithm. DoS have minimum score for detection, probe and R2L have the highest scores for detection.

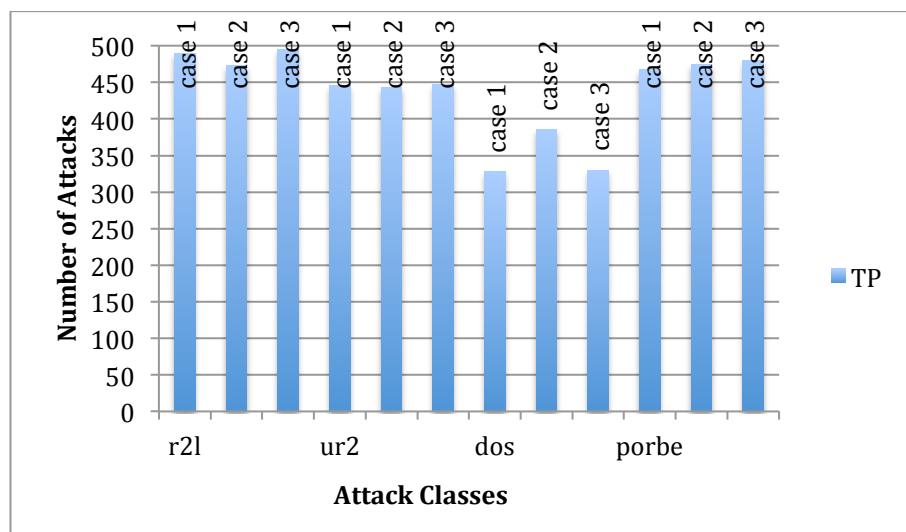


Figure 20 Attacks detected with DenStream

Table 11 has the output result values for TPR and FPR from running DenStream algorithm. The elapsed time for running DenStream is about 25s.

Table 11 Output Results of TPR and FPR

Attack	TPR	FPR
R2L	0.980	0.280
	0.946	0.304
	0.990	0.350
U2R	0.892	0.280
	0.886	0.304
	0.894	0.350
DoS	0.656	0.280
	0.772	0.304
	0.660	0.350
Probe	0.963	0.280
	0.946	0.304
	0.976	0.350

Fig.21 also illustrates the results of TPR and FPR of all cases for the DenStream algorithm.

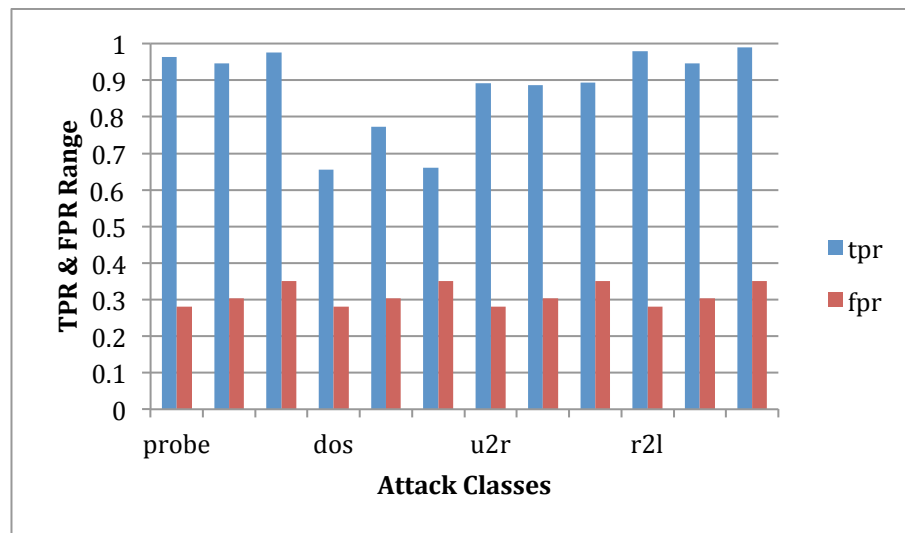


Figure 21 TPR and FPR of DenStream

Finally, The employment of DenStream algorithm in single run to get all results has been presented in 4.1. As parameter space is scanned to find the most appropriate parameter set, DBSCAN experiments cost more time because of running repetitively

to attain the best output results. But the time cost duration for detecting anomalous with DenStream measured is 25s for all attack types.

4.3 Experiments Over 2000 Packets

In this Section, in addition to the results given in previous section, we have randomly taken 1000 packets training and 1000 packets testing, using four types of attacks category (R2L, U2R, DoS and probe). In this part, we present two charts plotted by using data from the output results of F1 score. Additionally each chart has two different cases. The first is by training 1000 packets for training with 1000 packets for testing randomly. The second is by training 2000 packets for training and 2000 packets for testing randomly. Overall the result is shown in Fig. 22 for DBSCAN algorithm.

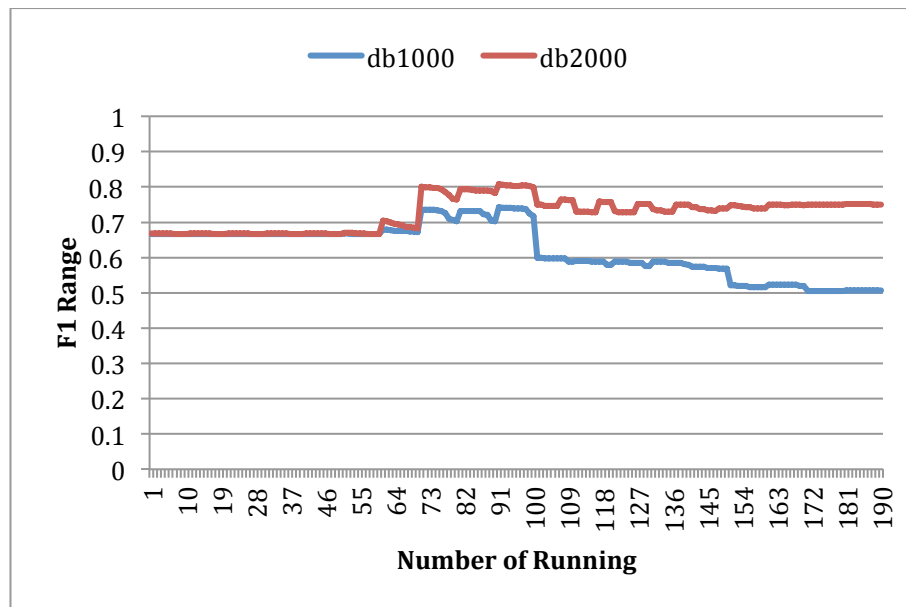


Figure 22 F1 score result for DBSCAN

Fig. 23 shows the output result for F1 score using two different results. First one is by training 1000 packets. Second one is testing 1000 packets randomly. Second, training 2000 packets and testing 2000 packets randomly. By using DenStream algorithm, data having any specific attack type can detect in this test.

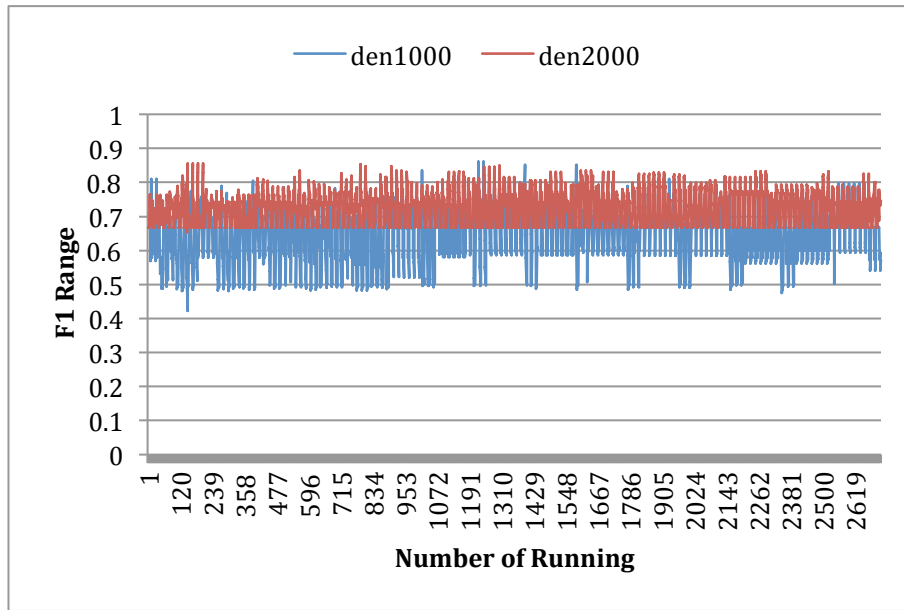


Figure 23 F1 score result for DenStream

CHAPTER 5

DISCUSSION

This chapter presents the overall results of this study by comparing the output results of two algorithms used for anomalous network packet detection according to F1 score, sensitivity, and number of attack detected and time of running.

5.1 F1 Score for DBSCAN and DenStream

According to F1 scores it can be judged that DenStream algorithm works better than DBSCAN. The value of F1 for the probe attack has the value 0.87 with DenStream. This is higher than 0.8 by using DBSCAN, for R2L, the value of F1 is 0.82, by using DenStream that is higher than the value by using DBSCAN, which was 0.75, also the value of F1 for the U2R attack has the value 0.75 by using DenStream, that is higher than 0.65 of DBSCAN, Finally the value of F1 for DoS attack has the value 0.65 with DenStream, which is higher than 0.60 that was obtained by using DBSCAN. It is at it minimum value detection with DenStream. Overall the DenStream has best results for R2L and probe as shown in Fig. 24.

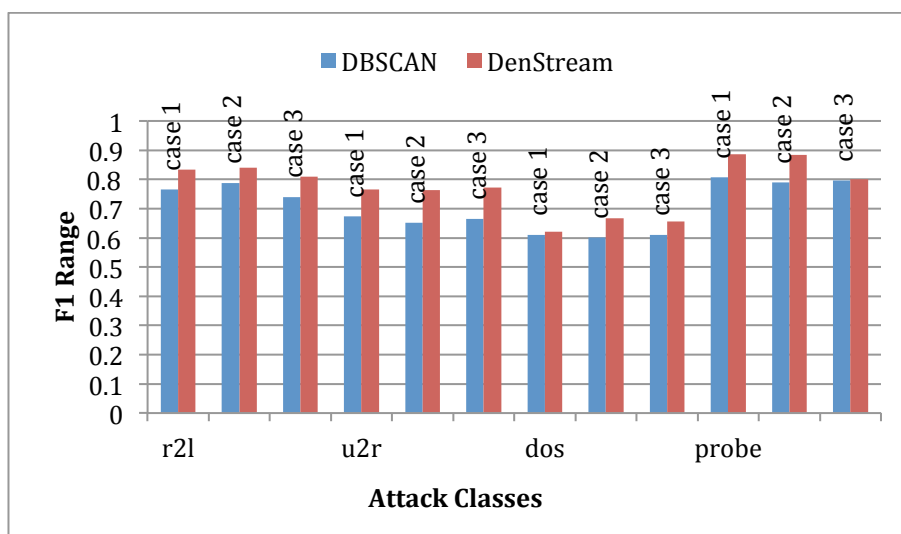


Figure 24 F1 score for both algorithms

5.2 Sensitivity of DBSCAN and DenStream

According to the output results of TPR, it can be realized that DenStream algorithm works acceptably and efficiently. The sensitivity rate value of DBSCAN for the probe attack has value approximately 0.98 with DenStream. That is the highest rate of TPR range. It is higher than 0.8 with DBSCAN, the value of TPR for R2L attack has values around 0.98 with DenStream that is higher than 0.78 using DBSCAN, the value of TPR for U2R attack has the result approximately 0.90 with DenStream. That is the highest rate of TPR range. It is also higher than 0.78 with DBSCAN. Last, the sensitivity rate value of TPR for DoS attack has value about 0.70 with DenStream. This is the highest rate of TPR range. It is higher than 0.6 using DBSCAN but it is at the lowest value for detection. Over all the DenStream has the best output value with R2L and probe, and R2L is worse but relatively better than DoS. The results have been given in Fig. 25.

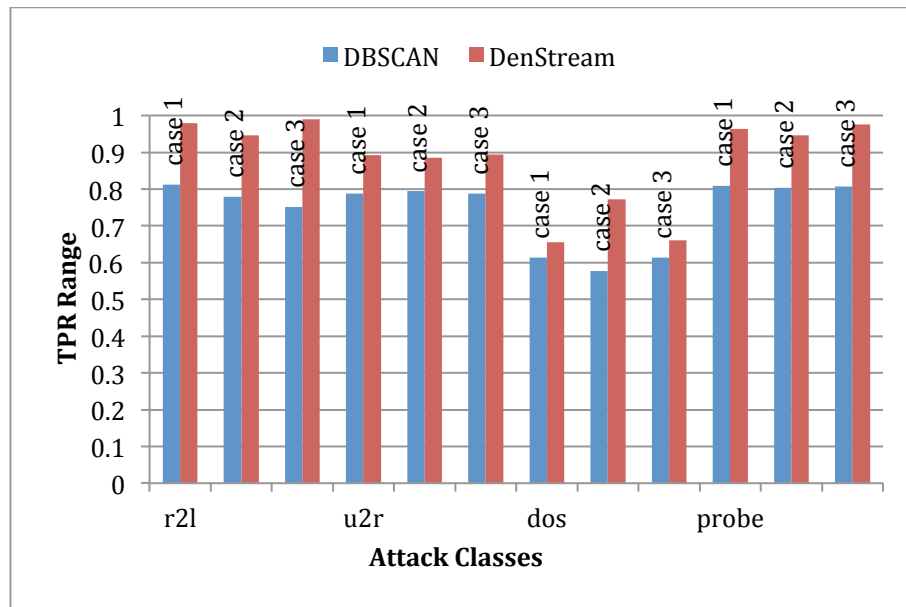


Figure 25 TPR for both algorithms

5.3 FPR for DBSCAN and DenStream

The output results of FPR indicate that DBSCAN algorithm has values lower than DenStream regarding the number of normal packets identified. For R2L has the value

0.25 normal packets that is less than the value 0.35 for DenStream, overall the approximate value for normal packets detection has maximum value around 0.35 with DenStream, and around 0.3 for DBSCAN and minimum value is about 0.3 with DenStream and 0.2 with DBSCAN. It is given in Fig. 26.

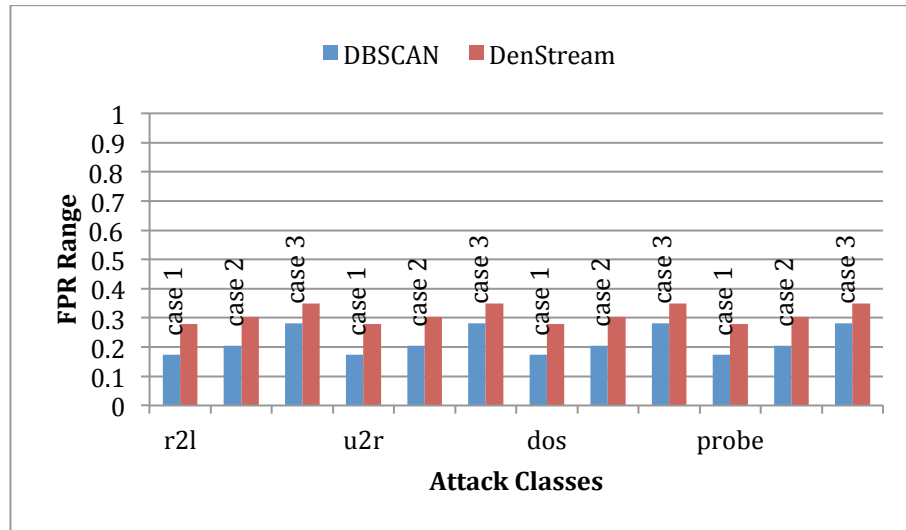


Figure 26 FPR for both algorithms

5.4 Attack Detection for DBSCAN and DenStream

The output results of TP show that DenStream algorithm can detect attackers efficiently and better than DBSCAN for the number of attacks identified. For R2L attack, it has the value about 500 attacks that is greater than the approximate value, which was 400 using DBSCAN. The number of attacks identified for probe attack has the value about 470 attacks, which is greater than the value 380 attacks that uses DBSCAN, the number of attacks identified for U2R attack has the value 440 attacks, that is greater than 390 attacks with DBSCAN, the number of attacks identified for DoS attack has 360 attacks, this is also greater than 270 attacks obtained with DBSCAN, Over all the detection numbers of attacks with DenStream algorithm is greater than the detection numbers of attacks with DBSCAN algorithm as shown in Fig. 27.

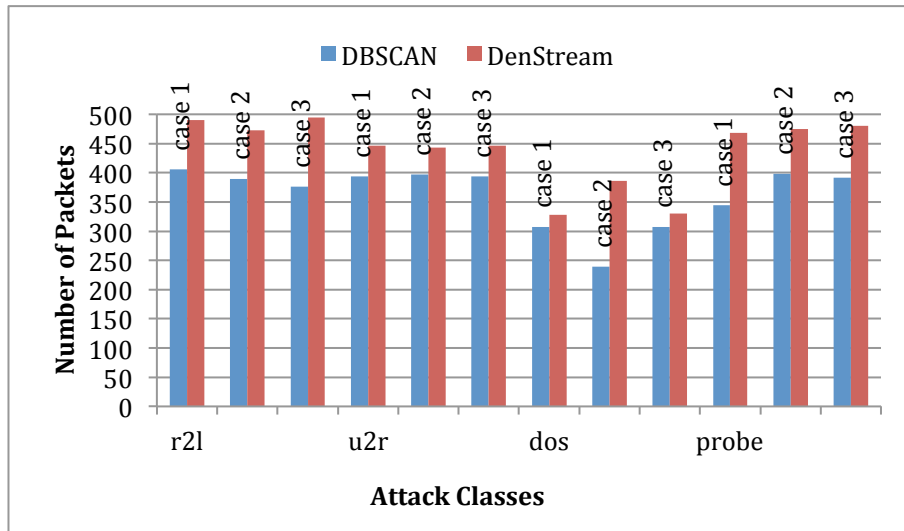


Figure 27 Number of attacks detected for both algorithms

5.5 Running Time to Get Result

The runtimes of algorithms in each output results in single run has been given in Fig. 28. It can be evaluated that DenStream algorithm can detect attackers faster than DBSCAN regarding the running time of detecting attacks. The running time average for DBSCAN is about 96s, but DenStream algorithm has very low running time average to detect attacks at about 25s, that makes DenStream faster and more efficient than DBSCAN for anomalous network packet detection in this study.

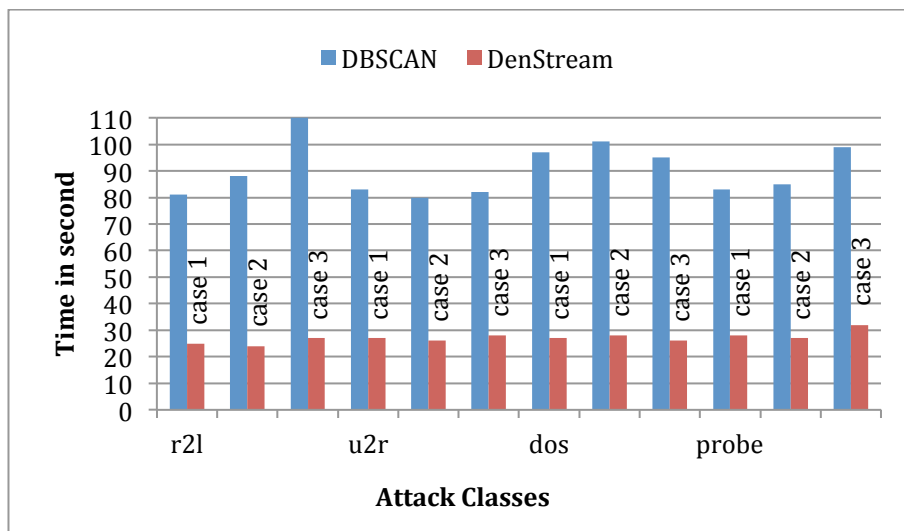


Figure 28 Running time for both algorithms

Furthermore, we can evaluate the run time of a single run for each algorithm DBSCAN and DenStream. Their results have been clearly given in Table 12. It summarizes elapsed time required for the detection of each attack types in seconds.

Table 12 Detecting Process Time

Attack	DBSCAN	DenStream
R2L	81s	25s
	88s	24s
	110s	27s
U2R	83s	27s
	80s	26s
	82s	28s
DoS	97s	27s
	101s	28s
	95s	26s
Probe	83s	28s
	85s	27s
	99s	32s

Another run time comparison has been presented in this section. One case has been taken and we have tested the values in many test ranges to improve the best-run time. So for both algorithms DBSCAN and DenStream training 1000 packets and testing 1000, 2000, 3000, 4000, 5000 and 10000 packets have been used for measuring running time for detecting anomalous network packets. The running time for getting the output has been given in Table 13. Time is measured in unit of seconds.

Table 13 Running Time Both Algorithms

Test	DB Time	Den Time
1000	106s	25s
2000	163s	27s
3000	220s	29s
4000	282s	31s
5000	321s	31s
10000	600s	42s

According to the Table 13 DenStream algorithm works faster than DBSCAN algorithm while classifying data for detecting anomalous network packets. The difference can be observed in Fig. 29 clearly. Experiments have been conducted with 1000, 2000, 3000, 4000, 5000 and 10000 packets testing. Elapsed time has been given in unit of seconds.

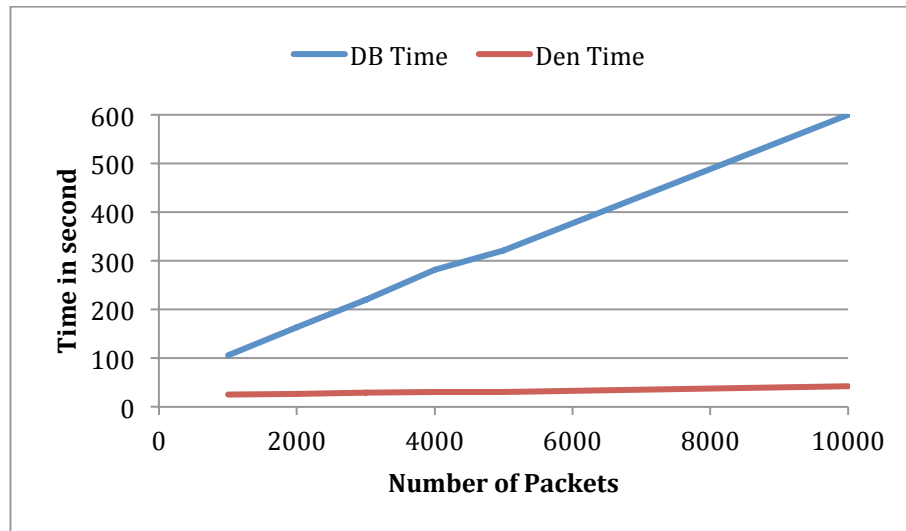


Figure 29 Running time comparison

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we applied two density base clustering algorithms for data stream mining techniques to detect anomalous from network traffic stream data. For this study we used a public and high profile dataset DARPA'99. Firstly, stream mining density clustering DBSCAN algorithm was used to extract 1-gram features and gather the output results but the sensitivity was not very high. We also needed a long time to detect anomalous. Secondly, other algorithms of density clustering used DenStream for detecting anomalous based on DBSCAN initializing on the same dataset by taking the same amount of packets using this algorithm, so the output result was more efficient and active and with a higher sensitivity and costing less time that with DBSCAN algorithm.

Table 14 Sensitivity Higher Results

	R2L	U2R	DoS	Probe
DBSCAN	81%	79%	61%	80%
DenStream	99%	89%	77%	98%

This study reached a best result for detecting algorithm based on density as 99% sensitivity for DenStream, otherwise 81% for DBSCAN. Also, the lower rate was 77% for DenStream, otherwise 61% for DBSCAN according to the Table 14. Moreover, the best result reached by this work for F1 score value for DenStream is 87% otherwise the F1 score value for DBSCAN is 75%. Furthermore, time is critical for detection attackers or anomalous network packets, so the total time cost for DBSCAN for detecting about 90s for each running time and DenStream takes less than about 25s for each running time. This study noted that DenStream could work as an efficient and effective algorithm in data stream mining techniques, for detecting anomalous of network attacks. Also, the advantage of this algorithm is how we can

save time and reduce memory and process use for analyzing and detection. According to Table 14 DenStream algorithm can detect the high volume of packets in a short time when comparing it with DBSCAN running time. The time is important because the internet is constantly speeding up the detection techniques must also be faster.

In the future, we can use these techniques in a real-time method for detection attacks. Moreover, use it as a part of a system to prevent them, or can be used as a part of anti-virus applications or in malicious code detection in real time. Lastly, it can be used in detection engine for intrusion detection in any stream data using n-gram method for attack detection also.

REFERENCES

1. **Miller Z., Deitrick W., Hu W., (2011)**, “*Anomalous Network Packet Detection Using Data Stream Mining*”, Journal of Information Security, California, vol. 2, pp.158-168.
2. **Suchulman D., (2014)**, “*Internet Security Threat Report*”, Symantec Corporation, California, vol. 19, pp. 5-60.
3. **IT Security Threats Symantec**, “http://www.symantec.com/security_response/”, (Data Download Date: 23.06.2014).
4. **Suchulman D., (2013)**, “*Internet Security Threat Report*”, Symantec Corporation, California, vol. 18, pp. 6-30.
5. **Lin D., (2013)**, “*Network Intrusion Detection and Mitigation against Denial of Service Attack*”, Master Thesis, University of Pennsylvania, pp. 10-15.
6. **Miller Z., Hu W., (2012)**, “*Data Stream Subspace Clustering for Anomalous Network Packet Detection*”, Journal of Information Security, California, vol. 3, pp. 215-223.
7. **Anand A., Patel B., (2012)**, “*An Overview on Intrusion Detection System and Types of Attacks It Can Detect Considering Different Protocols*”, International Journal of Advanced Research in Computer Science and Software Engineering, India, vol. 2, pp. 94-98.
8. **Mahoney M. V., (2003)**, “*Network Traffic Anomaly Detection Based on Packet Bytes*”, Proceeding SAC '03 Proceedings of the 2003 ACM Symposium on Applied Computing, New York, pp. 346-350.

9. **Faizal M. A., Mohd Z. M., Sahib S., (2010)**, “*Time Based Intrusion Detection on Fast Attack for Network Intrusion Detection System*”, Second International Conference on Network Applications, Protocols and Services, Kedah, Malaysia, pp. 148-152.
10. **Oza A., Ross K., Low R. M., Stamp M., (2014)**, “*HTTP Attack Detection Using n-gram Analysis*”, Computers & Security, Elsevier, Massachusetts, vol. 45, pp. 242-254.
11. **Wang K., Cretu G., Stolfo S. J., (2006)**, “*Anomalous Payload-Based Worm Detection and Signature Generation*”, Recent Advances in Intrusion Detection RAID, Springer Science & Business Media, Berlin, vol. 3858, pp. 227-246.
12. **Perdisci R., Ariu D., Fogla P., (2009)**, “*McPAD: A Multiple Classifier System for Accurate Payload-Based Anomaly Detection*”, Computer Networks, Elsevier, Massachusetts, vol. 53, pp. 864–881.
13. **Mahoney M. V., Chan P. K., (2002)**, “*Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks*”, The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, pp. 376-385.
14. **Ingham K. L., Inoue H., (2007)**, “*Comparing Anomaly Detection Techniques for HTTP*”, Recent Advances in Intrusion Detection RAID, Springer Science & Business Media, Berlin, vol. 4637, pp. 42-62.
15. **Han J., Kamber M., (2006)**, “*Data Mining: Concepts and Techniques*”, Elsevier, California, no. 2, pp. 7-41.
16. **Witten I. H., Frank E., Hall M. A., (2011)**, “*Data Mining Practical Machine Learning Tools and Techniques*”, Elsevier, Massachusetts, no. 3, pp. 3-29.
17. **Han J., Kamber M., Pei J., (2012)**, “*Data Mining Concepts and Techniques*”, Elsevier, Massachusetts, no. 3, pp. 6-33.
18. **Gama J., (2010)**, “*Knowledge Discovery from Data Streams*”, Taylor & Francis, New York, pp. 79-97.

19. **Dean J., (2014)**, “*Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners*”, John Wiley & Sons, Indiana, pp. 127-160.
20. **Maimon O., Rokach L., (2010)**, “*Data Mining and Knowledge Discovery Handbook*”, Springer Science & Business Media, New York, no. 2, pp. 259-287.
21. **Gama J., (2010)**, “*Knowledge Discovery from Data Streams*”, Taylor & Francis, New York, pp. 7-31.
22. **Nagabhushana S., (2006)**, “*Data Warehousing OLAP and Data Mining*”, New Age International, New Delhi, vol. 86, pp. 251-260.
23. **Aggarwal C. C., (2007)**, “*Data Streams: Models and Algorithms*”, Springer Science & Business Media, New York, pp. 289-298.
24. **Han J., Kamber M., (2006)**, “*Data Mining: Concepts and Techniques*”, Elsevier, California, no. 2, pp. 383-466.
25. **Han J., Kamber M., Pei J., (2012)**, “*Data Mining Concepts and Techniques*”, Elsevier, Massachusetts, no. 3, pp. 443-490.
26. **Bell J., (2014)**, “*Machine Learning: Hands-On for Developers and Technical Professionals*”, John Wiley & Sons, Indiana, pp. 275-310.
27. **Cole E., (2002)**, “*Hackers Beware*”, Sams Publishing, Indiana, pp. 605-680.
28. **Information Security, Network Security, System Security Tutorials and Study Materials**, “<http://www.omniseu.com/security/>”, (Data Download Date: 15.09.2014).
29. **Anderson R. J., (2010)**, “*Security Engineering: A Guide to Building Dependable Distributed Systems*”, John Wiley & Sons, Indiana, no. 2, pp. 633-678.

30. **Anderson R. J., (2001)**, “*Security Engineering: A Guide to Building Dependable Distributed Systems*”, John Wiley & Sons, Indiana, no. 1, pp. 367-390.
31. **Security Engineering: A Guide to Building Dependable Distributed Systems**, “<http://www.cl.cam.ac.uk/~rja14/book.html>”, (Data Download Date: 20.08.2014).
32. **Corona I., Giacinto G., Roli F., (2013)**, “*Adversarial Attacks Against Intrusion Detection Systems: Taxonomy, Solutions and Open Issues*”, Journal Information Sciences: an International Journal, Elsevier, Massachusetts, vol. 239, pp. 201-225.
33. **Northcutt S., Novak J., (2002)**, “*Network Intrusion Detection*”, Sams Publishing, Indiana, no.3, pp. 125-203.
34. **Ariu D., Tronci R., Giacinto G., (2011)**, “*HMMPayl: an Intrusion Detection System Based on Hidden Markov Models*”, Computers & Security 2011 Export: BIBTEX LNCS IEEE ACM, New York, vol. 30, pp. 221-241.
35. **Pathan K. S., (2014)**, “*The State of the Art in Intrusion Prevention and Detection*”, CRC Press, Florida, pp. 3-45.
36. **Aziz A. S., Salama M., Hassanien A., (2012)**, “*Detectors Generation using Genetic Algorithm for a Negative Selection Inspired Anomaly Network Intrusion Detection System*”, Proceedings of the Federated Conference on Computer Science and Information Systems, IEEE, Wroclaw, pp. 597-602.
37. **Portillo S. P., (2014)**, “*Attacks Against Intrusion Detection Networks: Evasion, Reverse Engineering and Optimal Countermeasures*”, Doctoral Thesis, University of Carlos III, Madrid, pp. 25-35.
38. **Fung C., Boutaba R., (2013)**, “*Intrusion Detection Networks: A Key to Collaborative Security*”, CRC Press, Florida, pp. 21-37.
39. **Pastrana S., Gimenez C. T., Nguyen H. T., Orfila A., (2015)**, “*Anomalous Web Payload Detection: Evaluating the Resilience of 1-grams Based Classifiers*”, Intelligent Distributed Computing VIII Studies in Computational Intelligence, Springer Science & Business Media, New York, vol. 570, pp. 195-200.

40. **DARPA Intrusion Detection Evaluation Dataset**, "<http://www.ll.mit.edu/mission/communications/cyber/CSTcorporation/ideval/data/1999data.html>", (Data Download Date: 01.07.2014).
41. **List of Attacks in DARPA Dataset**, "http://www.ll.mit.edu/ideval/files/master_identifications.list", (Data Download Date: 14.03.2015).
42. **Jeya P. G., Ravichandran M., Ravichandran C. S., (2012)**, "*Efficient Classifier for R2L and U2R Attacks*", International Journal of Computer Applications, New York, vol. 45, no. 21, pp. 28-32.
43. **Tang J., (2011)**, "*An Algorithm for Streaming Clustering*", Examensarbete 30 HP, Uppsala, pp. 7-21.
44. **Wang K., Stolfo S. J., (2004)**, "*Anomalous Payload-Based Network Intrusion Detection*", Recent Advances in Intrusion Detection RAID, Springer Science & Business Media, New York, vol. 3224, pp. 203-222.
45. **Shaikh S., Khan A. P., Mahajan V. S., (2013)**, "*Implementation of DBSCAN Algorithm for Internet Traffic Classification*", International Journal of Computer Science and Information Technology Research (IJCSITR), India, vol. 1, pp. 25-32.
46. **Sharma L., Ramya K., (2013)**, "*A Review on Density Based Clustering Algorithms for Very Large Datasets*", International Journal of Emerging Technology and Advanced Engineering, India, vol. 3, pp. 398-403.
47. **Amini A., Saboohi H., Wah T. Y., Herawan T., (2014)**, "*A Fast Density-Based Clustering Algorithm for Real-Time Internet of Things Stream*", Scientific World Journal, Kuala Lumpur, pp. 3-10.
48. **Voigtlaender P., (2013)**, "*DenStream: Density-Based Stream Clustering Algorithm*", Pro Seminar Paper in RWTH, Aachen, pp. 2-6.
49. **Cao F., Estery M., Qian W., Zhou A., (2006)**, "*Density-Based Clustering over an Evolving Data Stream with Noise*", Sixth SIAM International Conference on Data Mining, Bethesda, pp. 2-6.

50. **Binary Classification Performance Test Formula**,
“http://en.wikipedia.org/wiki/Precision_and_recall”, (Data Download Date: 03.03.2015).

51. **Powers D. M., (2007)**, “*Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*”, Technical Report SIE-07-001, Adelaide, pp.3-10.

52. **Sensitivity and Specificity for Classification Test**,
“http://en.wikipedia.org/wiki/Sensitivity_and_specificity#Sensitivity”, (Data Download Date: 04.03.2015).

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: MOHAMMED, Ahmed

Date and Place of Birth: 25 April 1979, Kirkuk

Marital Status: Married

Phone: +90 537 497 2521

Email: ahmedburhan79@gmail.com



EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya Univ., Computer Engineering	2015
B.Sc.	Technical College Kirkuk, Software Engineering	2003
High School	Al Waleed High Schools, Kirkuk	1998

WORK EXPERIENCE

Year	Place	Enrollment
2003	Logic Computer Center	Trainer
2005	Computer Center / Kirkuk University	Technician
2009	CISCO Academy / Kirkuk University	Teacher
2010	Video Conference / Kirkuk University	Coordinator

FOREIN LANGUAGES

Arabic, English, Turkish.

HOBBIES

Football, Writing poetry and Collecting Stamps.