Anomaly Detection System by Mining Frequent Pattern using Data Mining Algorithm from Network Flow

A. R. Jakhale[1], G. A. Patil [2]

[1] *M.E Student, Computer Science & Engineering Department, D. Y. Patil college of Engineering and Technology, Kolhapur, Maharashtra, India*
[2] Associate Professor, *Computer Science & Engineering Department, D.Y.Patil college of Engineering and Technology , Kolhapur, Maharashtra, India,*

Abstract

In recent year technology in computer with the development of internet technology, intrusion detection system has become a universal focus on the world. Intrusion detection techniques can be categorized into signature detection and anomaly detection. Signature detection lacks to detect the newly invented attacks. Anomaly detection is an important dilemma that has been researched within diverse research areas and application domains. Number of anomaly detection systems is developed with a high false alarm rate; fail in real -time detection, poor in detection while network traffic is high. So, a safe, stable and efficient network is necessary for today's society. We address these issues, first, by proposing an anomaly detection method providing a data mining algorithm that overcomes the common drawbacks of anomaly detectors based on statistical analysis, second, by providing both a benchmark tool that compares the results from historical normal data. Sliding window model and clustering is used to reduce complexity. The results show that algorithms are effective and system can definitely identify a lot of potentially very information in time which is useful to detect network anomalies with high detection rate.

Keywords: *Signature Detection, Anomaly Detection, Sliding window model, Data mining algorithms, clustering.*

1. Introduction

In recent years need for protection from various attacks & malicious traffic that originate from the Internet has gained focus. Rapid detection of network traffic anomalies is most significant in providing a secure network infrastructure. So, monitoring the network globally and timely is critically important for network administrators to find the entire network running regularly. Intrusion detection techniques can be categorized into signature detection and anomaly detection [1]. Signature detection systems use patterns of well-known attacks of the system to match and identify known intrusions. The main advantage of the signature detection skilfully detects instances of known attacks. The main disadvantage is that it lacks the ability to detect the newly invented attacks [2]. Anomaly detection is performed by monitoring and detecting changes in the pattern of resource utilization or behavior of the system based on comparing the current observed network behavior and a known model of normal network behavior. Anomaly detection refers to the problem of finding patterns in data that do not conform to the expected behavior [3]. Unfortunately, some anomaly detection system not well suited for high speed real time network with huge data travelling across the network also false positive rate is little higher. The premier goal of anomaly detection is to identify anomalous activities in a system accurately and in a timely fashion.

1.1 Research Objective:

- To assess the current state-of-the art for anomaly detection systems and identify different algorithms, important and desirable theoretical properties of the different anomaly detection system.
- To formulate a strategic plan and to develop a system for guiding the new monitoring anomaly detection system for networking. To monitor network traffic in real time and judge whether the whole network works well or not. Understand the active events that happen frequently and may influence or even ruin the total network. Build anomaly detection algorithm that is to prepare models of normal behavior and detect any deviation from it. Demonstration of feasibility and validity of proposed system on real world data sets. Identify suitable performance measures for evaluating proposed system for anomaly detection in networking.

The system aspires to use the technique of mining frequent pattern from network traffic. A design of sliding window model to make sure the mining result novel and integrated. The system consists two main stages training & testing namely. In the training phase which analyzes normal traffic, makes profiles that are expected for that service. The system calculates frequent sequential pattern approach to make the profile, then develop a mining algorithm that contains simple frequent mining pattern, fast update algorithm, capturing 1-pattern mining algorithm, capturing multi-pattern mining algorithm. To minimize the complexity of profile comparisons, profiles are clustered together. In the testing phase system captures incoming traffic which is compared with normal profile to get suspicious packet.

The rest of the paper is structured as follows. Section 2 discusses the related work of the anomaly detection. Section 3 discusses the framework for the new anomaly detection system. Section 4 illustrates implementation details for anomaly detection system. Section 5 discusses performance and evaluation. Section 6 discusses the conclusion and future scope.

2. Research Background

Signature based systems Snort and Bro are clueless in case of novel attacks where the attack pattern is not matched with stored signatures. NIDES [2,4] operate in real time for continuous monitoring of user activity or could run in a batch mode for periodic analysis of the audit data. PHAD [5] and ALAD [6]. To detect anomalies, PHAD and ALAD use port numbers, TCP flags, and keywords found in the payload. In addition, SPADE [7], and NATE [8], compute statistical models for normal network traffic. NETAD [5], monitors the first 48 bytes of each IP packet header and creates different models based on each individual network protocol. PAYL [9] and McPAD [2, 10]. Both used n-grams, sequences of n consecutive bytes in a packet's payload, as features to represent packets. Haystack [11] is an example of a statistical anomaly-based intrusion detection system. It used both user and group-based anomaly detection strategies, and modelled system parameters as independent, Gaussian random variables.

The implementation of PCNDA [12] was based on CPP (Content based Payload Partitioning) technique which divides the payload into different partitions depending on the content of the payload. In case of data mining, Denning is the first person which uses data mining techniques to solve problems in network security [13]. In sequential pattern mining techniques, which mine frequently occurring ordered subsequences as patterns. [11] categorize anomaly detection algorithms into statistical, data-mining and machine learning based. Some supervised algorithms for detection of network anomalies has been implemented [14].

Unfortunately, some anomaly detection systems fail for high detection rate. Also, can't deal with the gradual or abrupt change in the real time data flows. Not well suited for high speed network with huge data travelling across the network. Proposed system which mainly focused to use data mining techniques for high speed & accurate detection of anomaly.

3. Main title Proposed Framework for anomaly packet detection system

3.1 Basic Definitions: In this section, we first introduce the relevant basic concepts of anomaly Detection.

3.1.1.*Definition 1 (Anomaly):* Anomalies [3] are patterns in data that do not conform to a well defined notion of normal behavior. Fig. 1 illustrates anomalies in a simple 2-dimensional data set. The data have two normal regions, *N*1 and *N*2, since most observations lie in these two regions. Points that are sufficiently far away from the regions, e.g., Points *o*1 and *o*2, in region *O*3, are anomalies.

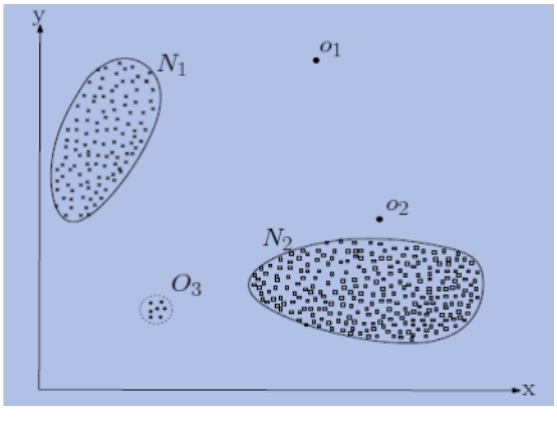


Fig.1. A simple example of anomalies in a 2-dimensional data set.

Assume that a training set X1 = {X, ...,X1} of nominal data samples is available. Given a test sample X, the objective of anomaly detection is to declare X to be an anomaly if X is significantly different from the samples in X1.

3.1.2.*Definition 2 (Anomaly Detection):* Anomaly detection [3] also referred to as outlier detection refers to detecting patterns in a given data set that do not conform to an established normal behavior.

3.1.3.*Definition 3 (Anomaly Detection Algorithms)*: Anomaly detection algorithms [13] build models of normal behavior and automatically detect any deviation from it. The major benefit of such algorithms is their ability to potentially detect unforeseen attacks.

3.2 Framework

The proposed framework aims to implement detection of all possible anomalies using mining by finding frequent sequential pattern. This is done by first analyzing packet of real network traffic. The layout of the proposed system is shown in Fig. 2.

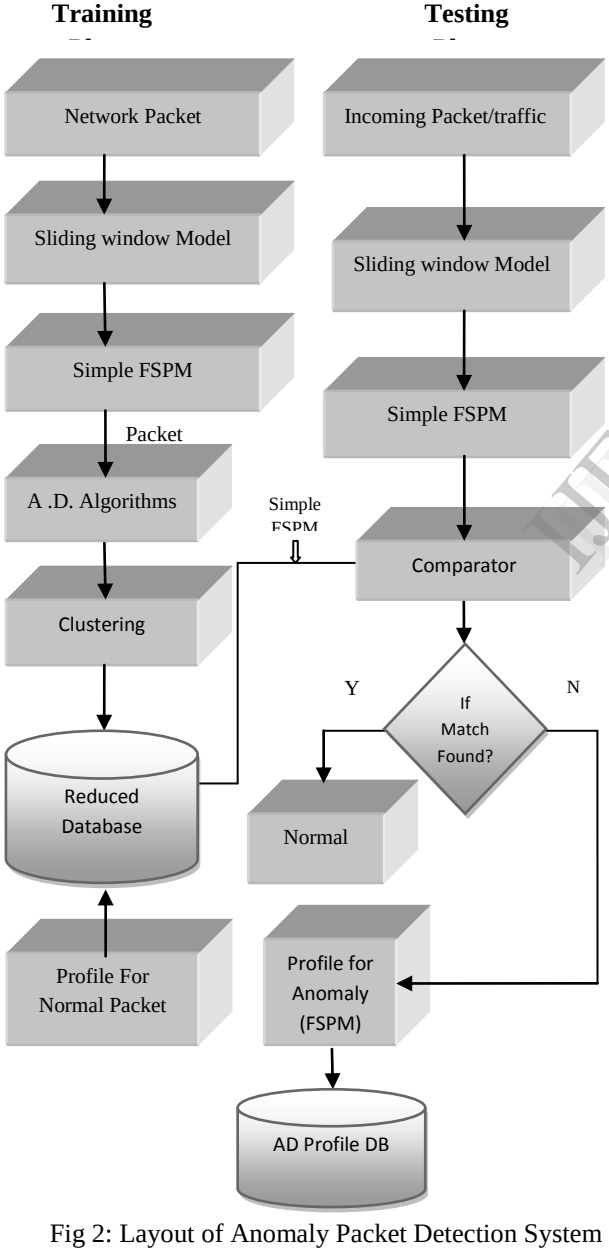


Fig 2: Layout of Anomaly Packet Detection System

3.2.1 Training Phase

- Capture Network Packet

  Packets are captured from a real network for selected features of packet during training and extract typical behavior from those and save as baselines and apply sliding window on those packets.

  Network flow is handled by a sliding window model. It uses a fixed length for flow control. The sliding window should contain all the fresh data in a network; so once a new piece of data appears, the sliding window should move forward to capture it. To avoid updating sliding window too frequently, we use basic windows with equal length (such as one minute or thirty second) to divide a sliding window into several continuous partitions shown in Fig 3.

- Create Profile for Captured traffic

  Since the infinite problem has turned into a finite problem with the sliding window model, we directly use the algorithm of simple frequent pattern mining method on the dataset within a sliding window [15-16].

Frequent sequential pattern (FSP): It is a set of items that occur frequently. **Support**: Support of frequent pattern is the times it occurs in the database (DB) [17]. **Tid-list**: Each transaction associated with its corresponding Tid-list. It is the set of all transaction IDs where it appears.

Below are the processing procedures.

- Simple Frequent Pattern Mining
  a. Scan the transaction database *DB*. Collect the set of frequent 1-patterns.
  b. Scan the transaction again to get Tid List of each frequent 1- pattern.
  c. To calculate the support of 2-Pattern ,we can simply intersect the two Tid-lists of corresponding 1-patterns composing that 2-pattern. Because the intersection contains the IDs of all transactions in which both the two 1-patterns occur. It is the same way to get all other frequent k-patterns (k>=2).

     Using this algorithm, we find set of frequent patterns. One move of sliding window brings lots of changes in frequent pattern so by using simple frequent mining algorithm we required so much scanning. The algorithm may be too long to satisfy online monitoring.
- Fast update mining algorithm

  Fast update mining algorithm is developed to avoid repeated scanning and mining in the sliding window[18].To mine frequent pattern from sliding window fast update mining algorithm first scans the new incoming basic window twice, use independent Tid_{BW}-list to calculate support instead of uniform

Tid_{BW}-list, and mines k-patterns by using the intersections of (k-1) -patterns' Tid-lists circularly (kP2). Fig.4 shows the mining structure based on the data in Table 1. It improves performance. Finding new emerging frequent patterns is complex, so we divide this problem into two sub-problems: capturing new frequent 1-patterns and capturing new frequent multi-patterns, and settle them separately in the following two sections.

- Capturing new frequent 1-pattern

Fast update mining algorithm t avoids the repeated scans and mining of the data in all overlapped basic windows. However, fast update mining algorithm fails to get the support of a frequent 1-pattern that is not in the pattern tree as not frequent in the last sliding window, but appears frequently in the new incoming basic window. The solution to this is use candidate queue. We put it in a queue as a candidate pattern, and record its corresponding Tid_{BW}-lists collected from that new incoming basic window. The next time another basic window comes, we check if it still keeps frequent locally on these two basic windows together. If not, we delete it from the candidate queue, else, record its new Tid_{BW}-lists collected from the second basic window. The same step is performed until when there are n record of Tid_{BW}-lists for that 1- pattern in the queue (n is the number of basic windows in a sliding window).

- Capturing multi-pattern algorithm

The multi-pattern re-mining algorithm is not effective enough from the algorithm point of view. So, still need to solve the problem of capturing new emerging multi-patterns which may be potentially frequent.

This algorithm procedure is given below.

a. Scans a small dataset in the new incoming basic window.
b. Fully uses the fast update mining algorithm to update the supported values of all known patterns in the pattern tree;
c. Use the candidate queue to capture new frequent 1-patterns and generate new frequent multi-patterns by the intersection of these new frequent 1-patterns' TidBW-lists, which is the key technique of the vertical mining method.

It gives the complete set of frequent pattern that we can use as a Profile of our captured traffic.

- Clustering

Clustering is used to group *similar* data instances into *clusters*. To minimize complexity of profile comparisons, profiles are clustered together. We have deployed k-means clustering to form a cluster, where we decide the number of clusters in advance and iteratively refine the midpoints of the clusters and group the closest points together with its appropriate midpoint of the cluster.

- Reduced Database

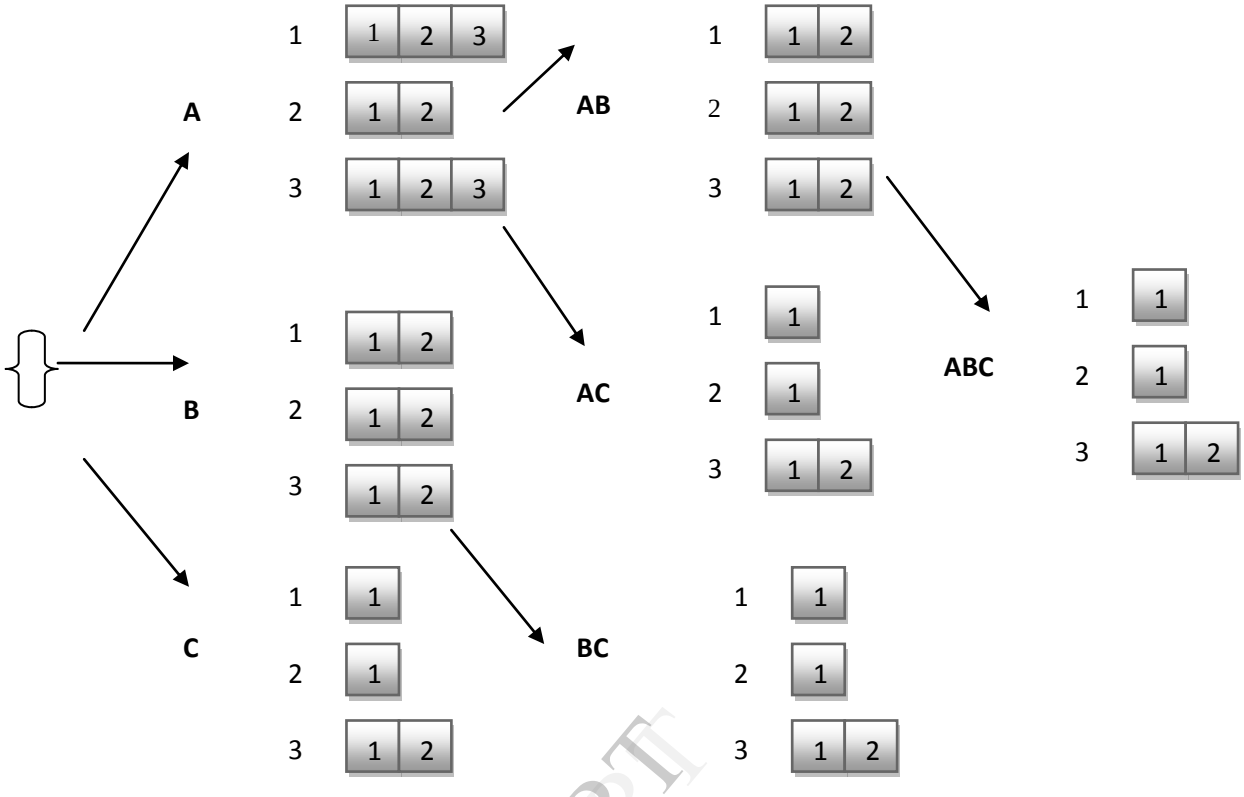
All above algorithms & clustering gives a profile for normal real time traffic.These profiles in a reduced database are used for further comparison in testing phase.

3.2.2. Testing Phase

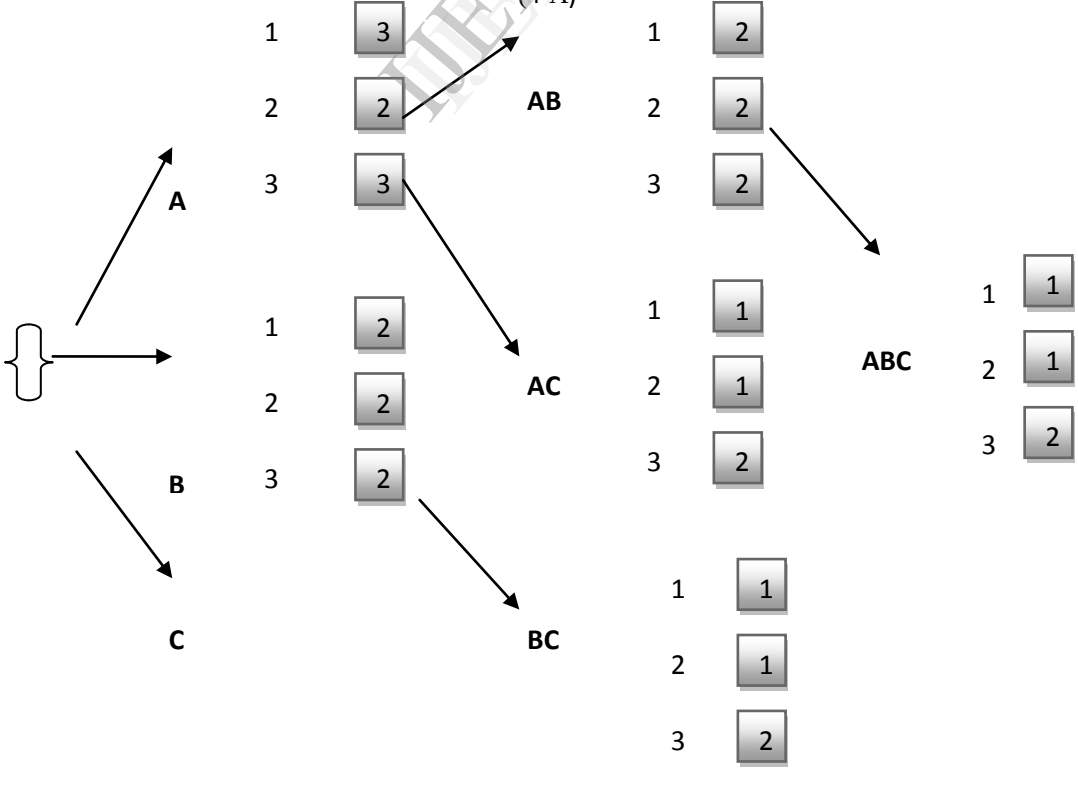
In the testing phase, the system captures incoming traffic. Create profiles for the same. Compares incoming traffic profiles with stored normal profiles which is in reduced database. If the new profile does not match with any stored profile for the same service, then an alert is generated indicating suspicious packet. Finally analyze the suspicious traffic characteristics of the baseline traffic.So this system not only provides a new application area for frequent pattern mining, but also provides a new technique for network monitoring.

Table 1: Transaction in A Sliding windows with three Basic windows.

ID_{BW}	ID_{SW}	Transaction
1	1	A B C
	2	A
	3	A B
2	4	B C
	5	A B C
	6	A
3	7	A
	8	A B C
	9	A B C



(4-A)



(4-B)

Fig 4: First running of Fast update mining and Pattern Tree.

4. Implementation Details

4.1 Tool Used

JDK1.7 is used for processing of packet information. JPCAP is a Java library which will enable us to capture packets and construct packets and process the packet information. When implementing the monitoring system in real time, we set the size of the sliding window 5 min and the size of basic window 1 min. So every minute the system performs the mining algorithm and updates the results one time. Mysql is used as a backend for reduced database.

The system requirements for the optimal execution of the code is a Pentium 4 machine with 512 MB RAM and fast Ethernet NIC(s).

4.2 Framework Generation

The collected TCPDUMP file is processed using the JPCAP and generated profile for real networks. Clustering is applied to profiles and obtained profiles saved in a database. Online traffic is compared to prepared profiles and finds whether incoming traffic is suspicious or normal.

5. Performance and Evaluation

The way the scheme has been conceived and implemented has several advantages over other similar efforts:

a. Anomaly detection based on expected behavior and the study of the above algorithm, guarantees a better longevity with respect to detection mechanisms based on pattern matching and signature detection. Our algorithms are effective enough to be used in real time network monitoring, especially the algorithms of fast frequent multi-pattern capturing algorithm and which dynamically produce and maintain the frequent patterns

b. Identifying attacks or malicious activity in a network with a high detection rate while maintaining a low false alarm rate is an aim of an anomaly detection system. Detection rate refers to the percentage of detecting attacks among all attack data. False alarm rate refers to the percentage of normal data which is recognized as an attack.
The system gives a detection rate is higher than the false alarm rate which is shown in Fig 5 according to Table 2.

Table 2: Detection rate and False Alarm rate

Packet Captured Time(s)	60	120	180
Detection rate(%)	0.7	0.73	1.8
False Alarm Rate(%)	0.1	0.2	0.3

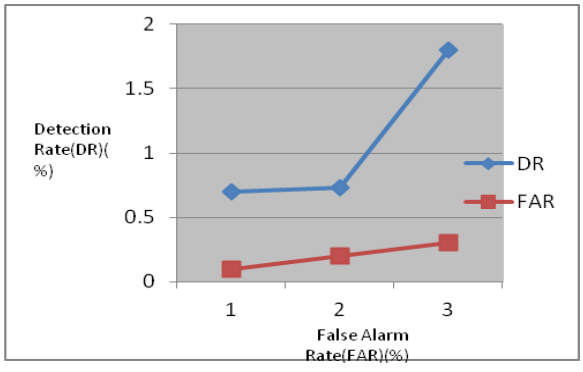


Fig 5: Detection Rate

c. The system is effective in real time traffic. It monitors the whole network to see whether it works well or not. It finds out attack For example, Distributed Denial of Service (DDoS) is one of dangerous Internet attacks, in which a multitude of compromised hosts attacks a single target, thereby causing denial of service for users of the targeted system.

d. The system is well suited for high speed network with huge data travelling across the network. Good results at ports like 22,23, and 25. The false positive rate is also low.

e. This study has highlighted that attacks, when classified in terms of anomaly categories, are very few with respect to the large number of signatures and patterns that similar solutions need to handle. This means that a System is much simpler to implement as the information that needs to be handled is very few.

f. This proposed system can definitely find out a lot of potentially very valuable information in time which greatly enhances the ability of campus network monitoring. The system can very well be integrated in an existing network environment. In particular it can feed network appliances such as the latest generation of border routers that allow network administrators to define thousand of network traffic rules per port, mostly produced by the System, with no performance glitch.

g. The study of the results produced by the ADS can be very well used for:

- Network bandwidth optimization
- Detection of network bandwidth killers
- Avoidance of unwanted protocols (e.g. Printers or proprietary protocols)

Network misconfiguration (e.g. Wrong DNS setup, usage of inexistent DHCP/ BOOTP servers); unwanted server activity detection (e.g. Installation by mistake of unwanted services);

6. Conclusion

In this paper, we propose an anomaly detection system. The system has two phases training and testing. In the training phase collect online network traffic which is processed using JPCAP. The profile is generated for network traffic by mining frequent pattern using algorithms Clustering is applied to reduce complexity in storage. In testing phase comparison is done with newly captured real traffic with normal traffic to detect anomalies. Our algorithms are effective enough to be used in real time network monitoring with high detection rate, especially the algorithms of fast frequent multi-pattern capturing algorithm. So the research in this paper is very valuable.

The system is well suited for high speed network with huge data travelling across the network. The false positive rate is little is also low. For further study data mining techniques can be applied in other domains such as data warehousing in order to improve the quality of data & exploiting a system interface to automatically reveal the patterns with which the administrator may be interested.

7. References

[1] Debar, H., M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion Detection Systems", *Computer Networks*, (1999), Vol.31, pp.805-822.

[2] Miller, Z., W. Deitrick, Hu Wei, "Anomalous Network Packet Detection Using Data Stream Mining", *Journal of Information Security*, (2011),Vol.2, pp.158-168.

[3] Chandola, V., A. Banerjee, and V. Kumar, "Anomaly Detection : A Survey" , *ACM Computing Surveys*, (2009), Vol.41, No.3 ,pp.1-58

[4] Andersan, D., T. Lunt, H. Havits, and A. Tamaru " Nides: Detecting unusual Program behaviour using statistical component of the next generation intrusion detection expert system", (1995), Technical report SRI-CSL-95-06.

[5] Mahoney, M., M. and P. Chan, "Learning non stationary models of normal network traffic for detecting novel attacks", *SIGKDD*, (2002), pp.376–385.

[6] Mahoney, M. "Network traffic anomaly detection based on packet bytes", *ACM-SAC, Melbourne*, (2003), pp. 346–350.

[7] Hoagland, J. "Spade. In Silican Defense", (2000), http: // www.silicondefense.com/ software /spice.

[8] Taylor, C., and J. Alves-Foss, "An empirical analysis of nate:Network analysis of anomalous traffic events", *New Security Paradigms Workshop*, (2002), pp.23-26

[9] Wang, K., and S. Stolfo, "Anomalous payload-based network intrusion detection" , *Recent Advances in Intrusion Detection*, (2004), pp. 203–222

[10] Perdisci, R., D. Ariu, "McPAD: A Multiple Classifier System for Accurate Payload-based Anomaly Detection",*Computer Networks*, *Special Issue on Traffic Classification and Its Applications to Modern Networks*, (2009), Vol. 5 , pp.864- 881.

[11] Patcha, A., and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest echnological trends", *Computer Networks*, (2007), Vol. 51 ,pp.3448–3470

[12] Thorat, S., A. Khandelwal, and K. Kishore, "Payload Content based Network Anomaly Detection", *Application of Digital Information and web Technologies*, (2008), pp. 127-132

[13] Denning, D. "An intrusion-detection model", *IEEE Transactions on Software Engineering*, (1987), Vol. 13, pp. 222–232.

[14] Sun, P., and S. Chawla, "On local spatial outliers", *IEEE International Conference on Data Mining*, (2004), Vol.19,pp.201-216

[15] Zaki, M., "Scalable algorithms for association mining" , *IEEE Transactions on Knowledge and Data Engineering*, (2000), Vol. 12 No. 3, pp.372–390.

[16] Zaki, M., and K. Gouda, "Fast vertical mining using diffsets", *International conference on knowledge discovery and data mining*, (2003), pp. 326–335.

[17] Li, X. and Z. Deng, "Mining frequent patterns from network flows for monitoring network" , *Expert Systems with Applications*, (2010), Vol. 37 ,pp.8850–8860

[18] Hong, T., C. Lin, and Y. Wu, "Incrementally fast updated frequent pattern trees" , *Expert Systems with Applications*, (2008), Vol. 34, pp. 2424–2435