



Clustering of Network Traffic by Online Streaming

Shital Salve*, Sanchika Bajpai, Sonali Khairnar, Bharati Patil

Computer Dept, Pune
India

Abstract— now days A large number of research is proposed on intrusion detection system, which leads to the implementation of agent based intelligent IDS (IIDS), Non – intelligent IDS (NIDS), signature based IDS etc. While developing such IDS models, learning algorithms from flow of network data plays a very important role in IDS systems. The proposed work focuses on implementing the new technique to cluster network data, traffic which eliminates the limitations in existing online clustering algorithms and prove the robustness and accuracy over large stream of network traffic arriving at extremely high rate. We are comparing the existing algorithm with new methods to analyse the accuracy and complexity.

Keywords— NIDS, IDS, Data Stream Mining, Online Clustering, RAH algorithm, Online Efficient Incremental Clustering algorithm, Ubiquitous mining

I. INTRODUCTION

Large amounts of data are emerging in the fields of network security, stock analysis, e-commerce and weather monitoring etc [1, 2]. These data are called data streams, which present new characteristics, such as fast, continuous, high-volume, open-ended, and concept drifting [3]. It is challenging for most of traditional algorithms for streaming data classification.

Fast growth in use of networking and internet makes security essential in recent years. Now days The most recent topic in network security is Network Intrusion Detection System (NIDS) which keeps the security at the highest level. Many diverse approaches have been proposed and implemented, which reduce the attacks and vulnerability in the network and makes it secure. Most widely used NIDS are signature based models [1]. Such models detect only known attacks, hence detecting unknown attacks without prior knowledge about specific intrusion remains a challenge. To cope with these challenges, intelligent IDS systems have developed [2]. The IIDS system focus on specific pattern of known attacks, which reveals the root cause of intrusion by constantly learning from network traffic, and if such patterns are identified and learned, they can produce the classification model for potential intrusion. Such systems are bundled with two layers, the first layer is training or learning layer, which learns the patterns of intrusion in the flow of network traffic. Another layer is testing, which applies learned rules to detect intrusions in unknown traffic data. As learning from online data is challenging than learning from static data, it became essential to provide attention towards accuracy of stream classification algorithms [3][4].

In this paper The proposed model focus on learning from network data by applying innovative stream clustering algorithms and then make use of produced clusters to build classification models for IIDS. Moreover the approach justify the efficiency and simplicity of new algorithm by comparing it with existing RAH clustering algorithm.

II. LITERATURE SURVEY

Data stream is an ordered sequence of points $A_1; \dots ; A_n$ that must be accessed in order and that can be read only once or a small number of times. Each reading of the sequence is called a linear scan or a pass. The stream model is motivated by emerging applications involving massive data sets; for example, customer click streams, telephone records, large sets of Web pages, multimedia data, financial transactions, credit card transaction, bank transaction and observational science data are better modelled as data streams. These data sets are far too large to fit in main memory and are typically stored in secondary storage devices Clustering, a useful and ubiquitous tool in data analysis, is, in broad strokes, the problem of finding a partition of a data set so that, under some definition of “similarity,” similar items are in the same part of the partition and different items are in different parts. The particular definition of clustering that is the focus of this paper is the k-Median objective, that of identifying k centers so that the sum of distances from each point to its nearest centre minimized.. The expansion of World Wide Web and increased use of internet has increased the risk of harmful intrusion every day. To cope with potential harmful intrusions, many diverse techniques have developed. The diverse approaches include histogram based anomaly detection models [5], Hidden Markow for IDS [6], IDS using Neural Networks [7], IDS using Genetic Algorithms [8] and Signature Based IDS [1].

NIDS using neural network introduces two layered architecture [7]. The first layer is training of neural network by either feed forward network or recurrent With the advance of sensor devices, many data are transmitted in the form of streams. Data streams are large in amount and potentially infinite, real time, rapidly changing, and unpredictable [3,4]. Compared with traditional data mining, ubiquitous data mining is more resource-constrained. Therefore, it may result in mining

failures when data streams arrive rapidly. Ubiquitous data stream mining thus has become one of the newest research topics in data mining [5].

Previous research on ubiquitous data stream clustering mainly adopts the AOG (Algorithm Output Granularity) approach [13], which minimize output granularity by merging clusters, so that the algorithm can adapt to available resources. Although the AOG approach can continue with mining under a resource-constrained environment, it sacrifices the accuracy of mining results. In this paper, we propose the new algorithm that can be used in mobile devices for clustering stream data. It adapts algorithm settings and compresses stream data based on currently available resources, so that mobile devices can continue with clustering at acceptable accuracy even under low memory resources.

Aggarwal et al. [11] proposed the CluStream clustering framework that consists of two components. The online component stores summary statistics of the data stream. The offline component uses summary statistics and user requirements as input, and utilizes an approach that combines micro-clustering with pyramidal time frame to clustering. The issue of ubiquitous data stream mining was first proposed by vivek et al. [8]. They analyzed problems and potential applications arising from mining stream data in mobile devices and proposed the LWC algorithm, which is an AOG-based clustering algorithm. LWC performs adaptation process at the output end and adjusts the minimum distance threshold between a data point and a cluster center based on currently available resources. When memory is full, it outputs the merged clustering results. Ching Ming [9] proposed RA-VFKM (Resource-Aware Very Fast K-Means), which borrows the effective stream clustering technique from the VFKM algorithm and utilizes the AOG resource-aware technique to solve the problem of mining failure with contained resources in VFKM. When the available free memory reaches a critical stage, it increases the value of allowable error (ϵ^*) and the value of probability for the allowable error (ϵ^*) to decrease the number of runs and the number of samples. Its strategy of increasing the value of error and probability compromises on the accuracy of the final results, but enables convergence and avoids execution failure in critical situations network and second layer introduces testing of network traffic by diverting it towards trained neural network. NIDS using data mining is most diverse among all approaches. The basic model introduces training and testing phases. The training phase learns the flow of network. To do so, it can use either online network stream or offline batch of network traffic data. To learn from network stream various stream classification algorithms are used, for e.g. CluStream [4], Hoeffding Tree and VFDT [3].

The signature based IDS systems uses attack signatures to classify unknown traffic, and updates signature data whenever new signatures are found. The data mining approach for NIDS also uses clustering approaches to group the network traffic in specific classes which can be further used by classification modules to classify the data with high accuracy. However the traffic is online and arriving at extremely high rate, which is to be clustered immediately when it arrives. This is concerned with online clustering algorithms and various online clustering approaches can be used to cluster this online data, but the issue remains is about the time complexity of online clustering algorithm. The time complexity is crucial part of such algorithm because the samples arrive so fast, and in large number so we would not have enough resources to store them before analysis. Moreover the clustering output is demanded very quickly by classification algorithms, where we cannot wait for batch of network packets to arrive which would be processed later.

To overcome above challenges resource aware high quality RAH-Clustering algorithm is implemented [9]. The algorithm computes available system memory and selects the upper bound of memory, data-centre threshold, centre - centre threshold and number of clusters. It then takes online samples of network traffic; cluster them into not more than k number of clusters. Next, it again computes the available and used memory and checks this value with upper bound of memory, if value is within the bound, it takes next samples to process, and otherwise it relaxes the values of data - centre threshold and centre - centre threshold. By doing so, the algorithm would require less amount of memory due to reduction of data samples to cluster. The data samples which are in the scope of above threshold values are clustered and out of scope values are thrown out of memory, hence preserving memory. However this algorithm has many pros and cons. The advantages are this algorithm compromises accuracy but never stops working. The disadvantages are- this algorithm requires initial number of cluster (k) values as well as data-data and data-centre threshold values as preliminaries.

Very Fast K-Means (VFKM) [13] is a clustering algorithm for data streams developed as a faster version of K-Means [7], to mine large quantities of continuous rapid streaming information. Our objective is to extend VFKM to a ubiquitous environment for execution on resource constrained devices. Using AOG. The significance of this work is that it enables VFKM to operate in ubiquitous environments.

III. IMPLEMENTATION DETAILS

The implementation is divided into three stages. The first stage is sniffing of network packets, the second stage is applying innovative online clustering algorithm and third stage is to compare existing clustering approach with new one.

The system architecture shows basic components and flow of working of the system. The packet snifer is responsible for collecting the network traffic, which can be configured to filter the traffic with specific attributes. The priority attributes are then selected and arranged in the increasing order. These samples are then applied to existing RAH clustering algorithm which assigns specific cluste to every individual sample. For RAH clustering algorithm, number of clusters k, is the prior input, which is major limitation of RAH while applying it to cluster network traffic, which is diverse in nature due to which number of clusters can not be judged before clustering the data.

Next, the same samples are applied to Online Efficient Incremental Clustering algorithm, where number of clusters, k is not the prior input. The clusters are formed according to diverse nature of traffic data. At the end we are comparing the results for accuracy and complexity of both the algorithms and justify that Online Efficient Incremental Clustering is best

suitable approach to cluster the network traffic. The basic component of system are packet sniffer, attribute and their priorities and clustering algorithms.

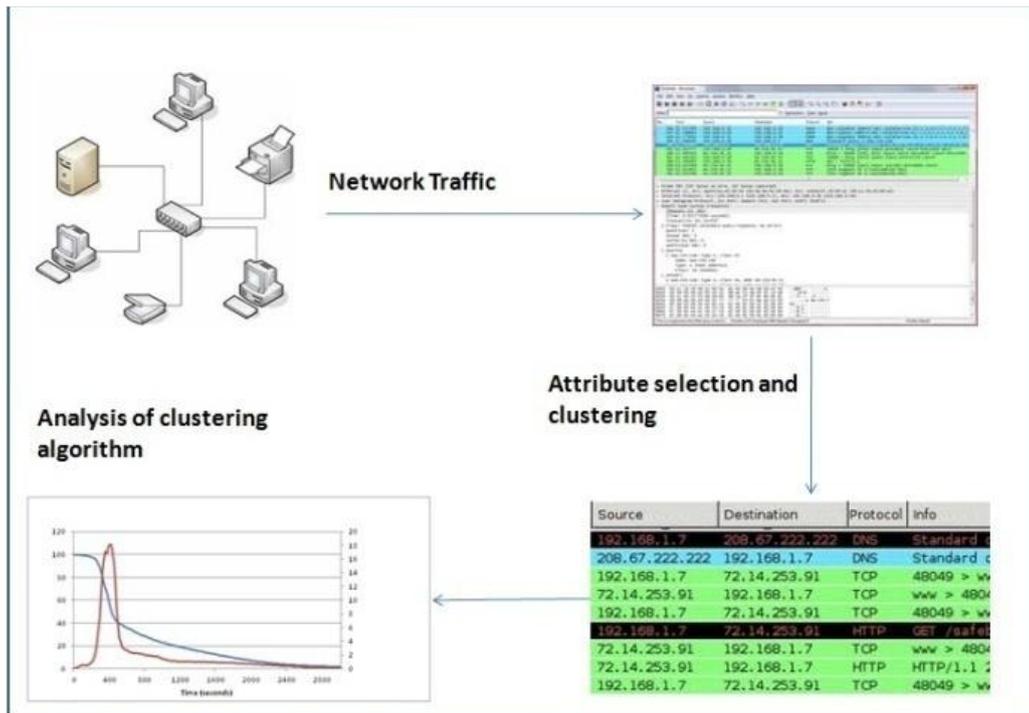


Fig. 1. Architecture of proposed system

A. How to sniff Packet

The packet sniffer sniffs the incoming packets through the network adapter. The sniffer is designed such that user can configure the attributes of packet that are traced by the sniffer. Many studies have developed that the attributes such as Source IP Address, Destination IP Address, Source Port Number, Destination Port Number, TCP Window Size, and TCP Data Length are most promising fields associated with different types of attacks [10], hence the above fields are considered while applying clustering algorithm on the stream of packets.

B. Attribute and their Priority Selection

Input to the clustering algorithm is mostly one or two dimensional numeric data points. By having single dimensional data, the similarity between two samples can be computed by taking direct difference between two values. For two dimensional sample data, the similarity between two samples is computed by Euclidian or Manhattan distance measures. But for multi-dimensional categorical data, the difference measure is very challenging. To calculate distance among network packets there is no standard measure.

The packet is set of attributes and every attribute may have numeric or categorical values. To compute the similarity among packets, first we need to focus on specific attribute values. If such selected attribute values for both packets are equal, then we can state that two packets are similar. But predicting such similarity on the basis of single attribute would not give accuracy, so we require multiple attributes and their priorities. The following algorithm explains the similarity measure between two packets based on attribute priority technique.

Input: $S1, S2, P1, P2, P3$

Output: $W=d(S1, S2)$

1. **Initialize** weight $(S1,S2)=0$;
2. **For each** attribute $S1_att$ in $S1$
3. **For each** attribute $S2_att$ in $S2$
4. **If** $S1_att = S2_att$
5. **If** priority of $S1_att = P1$ **then**
6. Increment the weight by weight factor=4
7. **Else if** priority of $S1_att = P2$ **then**
8. Increment the weight by weight factor=3
9. **Else if** priority of $S1_att = P3$ **then**
10. Increment the weight by weight factor=2
11. **return** weight

Fig. 2. Similarity measurement algorithm

The input to above algorithm is two packet samples and three attributes with increasing priorities. The algorithm compares every attribute of first sample with every attribute of second sample, if both attributes are equal, then it checks their priorities from available priority attributes. If the priority is highest, it increases the weight by weight factor 4, if priority is normal, then by 3 and if the priority is low then by 2. Finally it returns the weight.

C. Online Clustering

1) Resource Aware High Quality Clustering (RAHc)

Online learning algorithms require high efficiency by consuming very less amount of time and system resources. As the data arrives at extremely high rate, it is difficult to store it before analysis. The accuracy of online learner remained a big challenge in the fields of data mining.

To cope with the challenges of deploying the online learner on ubiquitous devices, there is need to work on resource awareness of learning algorithms. RAH clustering algorithm is one of them.

The algorithm computes available system memory and selects the upper bound of memory, data-centre threshold, centre - centre threshold and number of clusters. It then takes online samples of network traffic; cluster them into not more than k number of clusters. Next it computes the available and used memory and checks this value with upper bound of memory, if value is within the bound, it takes next samples to process, and otherwise it relaxes the values of data - centre threshold and centre - centre threshold. By doing so, the algorithm would require less amount of memory due to reduction of data samples to cluster. The data samples which are in the scope of above threshold values are clustered and out of scope values are thrown out of memory, hence preserving memory. However this algorithm has many pros and cons. The advantages are this algorithm compromises accuracy but never stops working. The disadvantages are- this algorithm requires initial number of cluster (k) values as well as data-data and data-centre threshold values as preliminaries. The algorithm is stated in figure 3.

Input: k, d, LB_m, x

Output: C

1. Compute N_m ;
2. $c \leftarrow \text{Random}(x)$;
3. **Repeat**
4. **For each** $x_i \in x$ **do**
5. **For each** $c_j \in c$ **do**
6. $D_i \leftarrow D \cup \{d^2(x_i, c_j)\}$;
7. **If** $\text{Min}[D_i] < \bar{d}$ **then**
8. $C_j \leftarrow C_j \cup \{x_i\}$ s.t. $d^2(x_i, c_j) = \text{Min}[D_i]$;
9. **Else**
10. delete x_i ;
11. Compute U_m ;
12. $R_m \leftarrow \frac{(N_m - U_m)}{N_m}$;
13. **If** $R_m < LB_m$ **then** $\bar{d} \leftarrow \bar{d} - \bar{d} \times (1 - R_m)$;
14. **If** $(1 - R_m) < 20\%$ **then** $\bar{d} \leftarrow \bar{d} + \bar{d} \times (R_m)$;
15. **For each** C_j **do**
16. $E \leftarrow \sum_{x_i \in C_j} \sqrt{(x_i - c_j)^2}$;
17. $S^2 \leftarrow \frac{(\sum_{x_i \in C_j} \sqrt{(x_i - c_j)^2})}{(\text{count}(C_j) - 1)}$;
18. $c'_j \leftarrow (\sum_{x_i \in C_j} x_i) / \text{count}(C_j)$;
19. **If** $c'_j \neq c_j$ **then**
20. $E' \leftarrow \sum_{x_i \in C_j} \sqrt{(x_i - c'_j)^2}$;
21. $\hat{S}^2 \leftarrow \frac{(\sum_{x_i \in C_j} \sqrt{(x_i - c'_j)^2})}{(\text{count}(C_j) - 1)}$;
22. **If** $(\hat{S}^2 < S^2)$ and $(E' < E)$ **then**
23. $c_j \leftarrow c'_j$;
24. **Else**
25. Output C_j ;
26. $\forall x_i \in C_j \quad x \leftarrow x - \{x_i\}$;
27. $c \leftarrow c - \{c_j\}$
28. **Else**
29. Output C_j ;
30. $\forall x_i \in C_j \quad x \leftarrow x - \{x_i\}$;
31. $c \leftarrow c - \{c_j\}$;
32. **Until** $\forall C_j (c'_j = c_j)$ or $(\hat{S}^2 \geq S^2)$ or $(E' \geq E)$
33. **Return**

Fig. 3. Resource Aware High Quality Clustering Algorithm

2) Online Efficient Incremental Clustering

Predefined number of cluster (k) and threshold values prior to online clustering are bottlenecks for online learner. The innovative Online Efficient Incremental Clustering algorithm copes with above bottlenecks and proves its ability to learn online without having predefined number of clusters (k) and any threshold values.

The algorithm initializes data – centre threshold (DC_{TH}) and centre – centre threshold (CC_{TH}) values as 0. It then read first available sample S. If S is the only sample in cluster space then the sample S would be the first member of new cluster. If S is not the only sample, then algorithm computes distance of sample S with centre of all available clusters. It then computes minimum distance D_i . Suppose the index of cluster to which S is having minimum distance is p.

By comparing sample S with all available centres, it yields three possible scenarios. In first scenario, the distance between sample S and cluster centre C[p] is 0. In this case sample S is merged into the cluster C[p]. The cluster centre of C[p] is updated and CC_{TH} also updated as minimum of available CC_{TH} . In second scenario the distance between sample S and cluster centre C[p] is greater than CC_{TH} . In this case the new cluster is formed having S as the member of that cluster. After that CC_{TH} is updated. In third scenario, the distance between sample S and C[p] is less than CC_{TH} . In this case sample S is merged into cluster C[p], CC_{TH} and DC_{TH} are updated. If DC_{TH} is greater than the CC_{TH} , then cluster C[p] is splitted to satisfy $CC_{TH} > DC_{TH}$.

Input: S

1. Initialize $DC_{TH} = 0$;
2. Initialize $CC_{TH} = 0$;
3. Read the sample S;
4. If S is the only sample Then
5. Initialize new cluster having S as cluster member;
6. Else
7. For each existing cluster i
8. For each cluster centre S_m of cluster i
9. Calculate $d_i = d(S, S_m)$
10. Initialize $D_i = \text{Min}\{d_i\}$;
11. If $D_i = 0$ Then
12. Merge S into cluster i ;
13. Update cluster centre for cluster i ;
14. Update CC_{TH} by selecting $\text{Min}\{D_{cc}\}$
15. Compute W_i
16. If $W_i > CC_{TH}$ Then
17. Split(cluster i);
18. Else if $D_i > CC_{TH}$ Then
19. Initialize new cluster having S as cluster member;
20. Update CC_{TH} by selecting $\text{Min}\{D_{cc}\}$;
21. Else if $D_i < CC_{TH}$ Then
22. Merge S into cluster i ;
23. Update CC_{TH} by selecting $\text{Min}\{D_{cc}\}$
24. Compute W_i
25. If $W_i > CC_{TH}$ Then
26. Split(cluster i);
27. Go to step 3

Fig. 4. Online Efficient Incremental Clustering

IV. MATHEMATICAL MODEL DESIGN

For our project we are going to use to how to generate micro clusters:

Step 1: Compute the mean of sampled data.

Step 2: Compute the square distance between the mean

and each data point, and find the data point nearest to the mean as a center point. Then move the window once to extract data.

Step 3: If the current number of center points is equal to user-defined number of micro-clusters q , execute Step 4; otherwise, return to Step 1.

Step 4: Use the K-Means algorithm to generate q micro-

clusters with q center points as cluster centroids, and compute the summary statistics of data points of each micro-cluster.

Assume that the user-defined number of micro-clusters is three. Table 2 shows the micro-clusters generated from the data points of Table 1, in which the micro-cluster Q1 contains three data points B (30, 21), D (24, 26), and H (21, 30) and the cluster feature vector is calculated as $\{(302 + 242 + 212, 212 + 262 + 302), (30 + 24 + 21, 21 + 26 + 30), 22 + 42 + 82, 2 + 4 + 8, 3\} = \{(1917, 2017), (75, 77), 84, 14, 3\}$.

Data point	Age	Salary (in thousands)	Arrival timestamp
A	36	34	t1
B	30	21	t2
C	44	38	t3
D	24	26	t4
E	35	27	t5
F	35	31	t6
G	48	40	t7
H	21	30	t8
I	50	44	t9

Table1: sample data

micro cluster	Data points	cluster
Q1	B,D,H	((1917,2017)(75,77) 84,14,3)
Q2	A,E,F	(3746,4846)(106,92)62,12,3)
Q3	C,G,I	((6740,4980)(142,122),139,1 9,3)

Table 2.clustering data incremental

V. RESULTS AND DATA SET

The implemented approach shows the network traffic captured by packet sniffer. The packet sniffer is configured to capture packets with attributes – source port, destination port, source IP address, destination IP address, TCP length, TCP checksum. In second window, the module clusters every network packet into specific category of cluster using RAH algorithm. For calculating similarity measurement among packet, three attributes are selected in their incremental priority order. These three attributes are source IP address, destination port number and TCP header length.

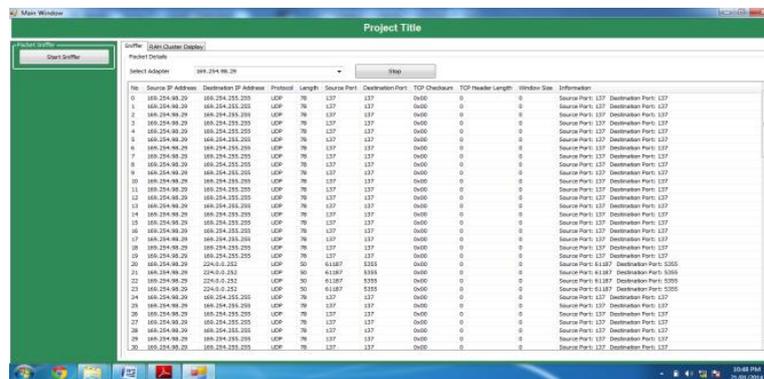


Fig. 5. Network data accepted by packet sniffer

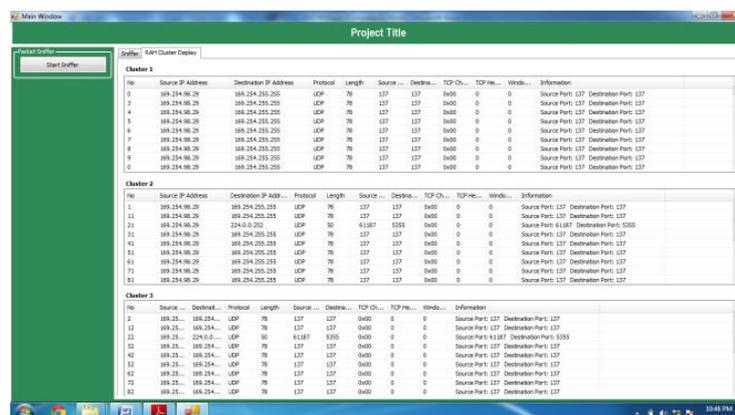


Fig. 6. Result of RAH clustering algorithm.

VI. CONCLUSION

The Online Efficient Incremental Clustering proves its effectiveness as it does not require the predefined number of cluster (k) and threshold values prior to the clustering. For clustering network data with extremely high rate, the above values are mostly unidentified and if stated would produce wrong results. The algorithm is best suited for such online clustering with high accuracy. Moreover the time complexity of resource aware learner is high due to the threshold bound checking and convergences of algorithm, which is completely eliminated in Online Efficient Incremental Clustering hence it is less complex than existing approaches.

REFERENCES

- [1] Farooq Anjum, Dhanant Subhadrabandhu and Saswati Sarkar, "Signature based Intrusion Detection for Wireless Ad-Hoc Networks: A Comparative study of various routing protocols", Telcordia. Tech Inc. Morristown NJ 07960J.
- [2] N.Jaisankar and R.Saravanan K. Durai Swamy, "Intelligent Intrusion Detection System Framework Using Mobile Agents", International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 2, July 2009R.
- [3] Pedro Domingos, Geoff Hulten, "Mining High Speed Data Streams", 2009
- [4] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, "A Framework for Clustering Evolving Data Streams", Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003.
- [5] Andreas Kind, Marc Ph. Stoecklin, and Xenofontas Dimitropoulos, "Histogram-Based Traffic Anomaly Detection", IEEE Transactions On Network Service Management, Vol. 6, No. 2, June 2009
- [6] Jiankun Hu and Xinghuo Yu, "A Simple and Efficient Hidden Markov Model Scheme for Host-Based Anomaly Intrusion Detection"
- [7] Jake Ryan, Meng-Jang Lin, "Intrusion Detection with Neural Networks", Advances in Neural Information Processing Systems 10, Cambridge, MA: MIT Press, 1998.
- [8] Vivek K. Kshirsagar, Sonali M. Tidke & Swati Vishnu, "Intrusion Detection System using Genetic Algorithm and Data Mining: An Overview", International Journal of Computer Science and Informatics ISSN (PRINT): 2231 – 5292, Vol-1, Iss-4, 2012.
- [9] Ching-Ming Chao and Guan-Lin Chao, "Resource-Aware High Quality Clustering in Ubiquitous Data Streams", in Proceedings of the 13th International Conference on Enterprise Information Systems, Beijing, China (2011).
- [10] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu, Aiko Pras and Burkhard Stiller, "An Overview of IP Flow-Based Intrusion Detection", IEEE Communications Surveys & Tutorials, Vol. 12, No. 3, Third Quarter 2010.
- [11] Aggarwal, C. C., Han, J., Wang, J. and Yu, P. S., "A Framework for Clustering Evolving Data Streams, Proceedings of the 29th International Conference on Very Large Data Bases, pp. 8192 (2003).
- [12] Qun Zhu, Xuegang Hu, Yuhong Zhang, Peipei Li, Xindong Wu "A Double-Window-based Classification Algorithm for Concept Drifting DataStreams" 2010 IEEE
- [13] "rahul shah, shonali krishnaswamay, mohammad gaber" "Resource aware very fast k means for ubiquitous data stream mining" 2005