ONLINE STREAM MINING APPROACH FOR CLUSTERING NETWORK TRAFFIC

Shital Salve[1], Sanchika Bajpai[2]

[1, 2]*Computer Department, Pune*

Abstract

A large number of research have been 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 building such IDS models, learning algorithms from flow of network traffic plays crucial role in accuracy of IDS systems. The proposed work focuses on implementing the novel method to cluster network 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 compare the existing algorithm with novel methods to analyse the accuracy and complexity.

Keywords— *NIDS, Data Stream Mining, Online Clustering, RAH algorithm, Online Efficient Incremental Clustering algorithm*

----------***----------

1. INTRODUCTION

Fast growth in use of networking and internet makes security essential in recent decades. 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 minimizes 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 evolved [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].

The proposed model focus on learning from network traffic 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.

2. LITERATURE SURVEY

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 evolved. 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 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.

3. 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.

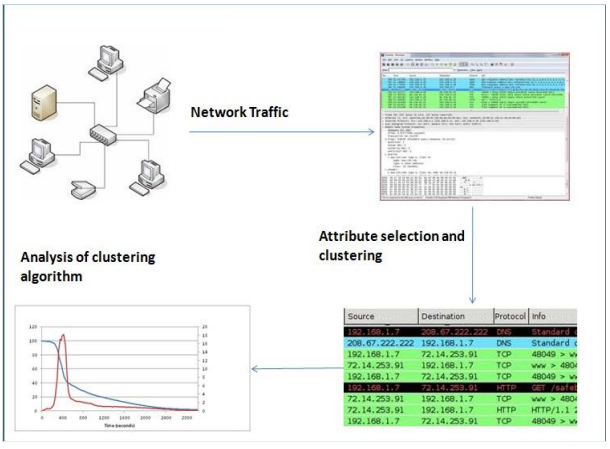

Fig 1 Architecture of proposed system

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 fromed 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 netwok traffic. The basic component of system are packet sniffer, attribute and their priorities and clustering algorithms.

3.1 Packet Sniffer

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 revealed 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.

3.2 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.

3.3 Online Clustering

3.3.1 Resource Aware High Quality Clustering (RAH)

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. $c \leftarrow Random\ (x)$;
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** $Min\ [D_i] < \bar{d}$ **then**
8. $C_j \leftarrow C_j \cup \{ x_i \}$ s.t. $d^2 (x_i , c_j) = Min\ [D_i]$;
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. $S^2 \leftarrow \frac{\left(\sum_{x_i \in C_j} \sqrt{(x_i - c_j)^2}\right)}{(count(C_j) - 1)}$;
18. $c_j^{'} \leftarrow (\sum_{x_i \in C_j} x_i) / count\ (C_j)$;
19. **If** $c_j^{'} \neq c_j$ **then**
20. $E^{'} \leftarrow \sum_{x_i \in C_j} \sqrt{(x_i - c_j^{'})^2}$;
21. $\hat{S}^2 \leftarrow \frac{\left(\sum_{x_i \in C_j} \sqrt{(x_i - c_j^{'})^2}\right)}{(count(C_j) - 1)}$;
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

3.3.2 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 Di. 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 spitted 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 *Sm* of *cluster i*
9. Calculate *di =d(S,Sm)*
10. Initialize *Di=Min{di};*
11. If *Di = 0* Then
12. Merge S into *cluster i*;
13. Update cluster centre for *cluster i*;
14. Update CC_{TH} by selecting *Min {Dcc}*
15. Compute *Wi*
16. If $Wi > CC_{TH}$ Then
17. Split(*cluster i*);
18. Else if $Di > CC_{TH}$ Then
19. Initialize new cluster having *S* as cluster member;
20. Update CC_{TH} by selecting *Min {Dcc};*
21. Else if $Di < CC_{TH}$ Then
22. Merge *S* into *cluster i*;
23. Update CC_{TH} by selecting *Min {Dcc}*
24. Compute *Wi*
25. If $Wi > CC_{TH}$ Then
26. Split(cluster *i*);
27. Go to step 3

Fig 4 Online Efficient Incremental Clustering

4. 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.

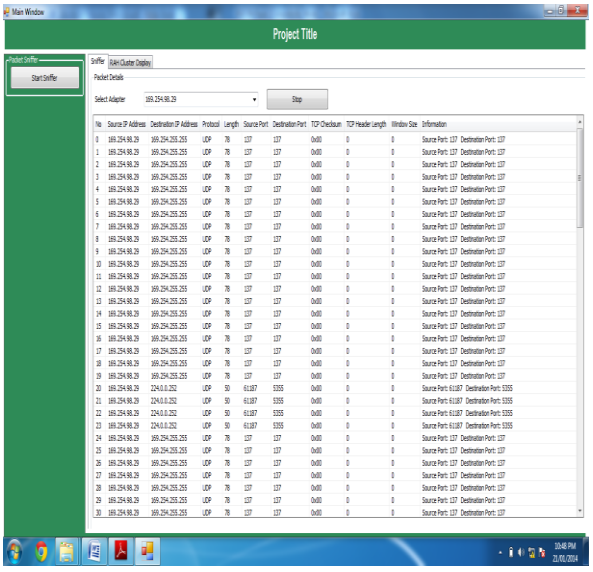

Fig 5 Network traffic captured by packet sniffer

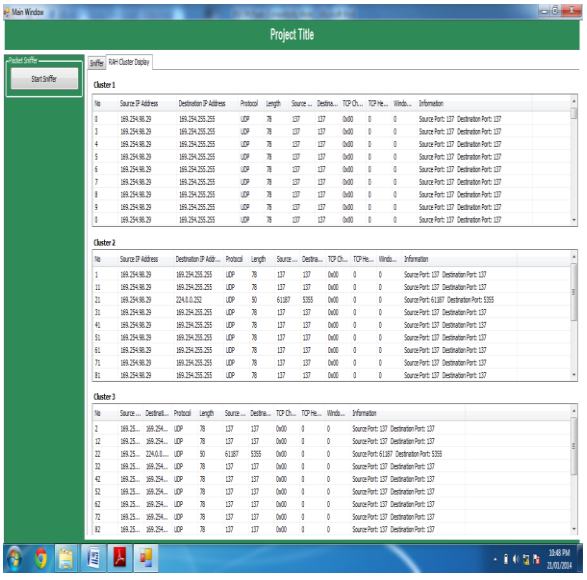

Fig 6 Three clusters of packets using RAH clustering algorithm.

5. CONCLUSIONS

The Online Efficient Incremental Clustering proves its effectiveness as it does not requires 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 unknown. 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".

[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.