ISSN:0975-9646

An Efficient ID3 Decision Tree for the Classification of Populated IP Address Using K-Mean Clustering

Rajesh Kumar Sharma[#], Anurag Jain[*], Subha Dubey[#]

[#]*CSE Department, RITS College*
RGPV University, Bhopal

[*]*HOD CSE Department*
RITS College BHopal

Abstract: In this paper addresses estimating the number of the users of a specific application behind IP address (IPs). This problem is central to combating abusive traffic, such as DDoS attacks, ad click fraud and email spam, scams, phishing, and malware distribution. Here we proposed an efficient method to classify the IP addresses that are associated with a large number of user requests. The idea is to classify the network traffic based on the IP addresses by first clustering the data using K-mean clustering and then applying horizontal partition based id3 decision tree.

1. INTRODUCTION

Online services such as Web-based email, search, and online social networks are becoming increasingly popular. While these services have become everyday essentials for billions of users, they are also heavily abused by attackers for nefarious activities such as spamming, phishing, and identity theft [1].

Simple conventional mechanisms for abuse detection that rely on source IPs set a limit, i.e., filtering threshold, on the IP activity within a time period. Once the limit is reached by an IP, either the IP traffic gets filtered for the rest of that time period, or the IP gets blacklisted for several consecutive periods. These techniques typically set the same threshold for all IPs. Setting an aggressive threshold yields a high false positive rate since some IPs have numerous users behind them and are hence expected to send relatively large traffic volumes. Setting a conservative threshold yields a high false negative rate, since the threshold becomes ineffective for distributed attacks where IPs send relatively little traffic. This work tailors the thresholds to the sizes of the IPs. It proposes a new framework for timely estimation of the number of users behind IPs with high enough accuracy to reduce false positives and with high enough coverage in the IP space to reduce false negatives [2].

Populated IP addresses (PIP) - IP addresses that are associated with a large number of user requests are important for online service providers to efficiently allocate resources and to detect attacks. While some PIPs serve legitimate users, many others are heavily abused by attackers to conduct malicious activities such as scams, phishing, and malware distribution. Unfortunately, commercial proxy lists like Quova have a low coverage of PIP addresses and offer little support for distinguishing good PIPs from abused ones [1].

On the one hand, not all proxies, NATs, or gateways are PIP addresses. Some may be very infrequently used and thus are not of interest to online service providers. On the other hand, while some PIP addresses may belong to proxies or big NATs, many others are not real proxies. Some are dial-up or mobile IPs that have high churn rates. Others include IP addresses from large services, such as Facebook that connects to Hotmail to obtain user email contacts. Additionally, not all PIPs are associated with a large number of actual users. Although many good PIPs like enterprise-level proxies are associated with a large number of actual users, some abused PIPs may be associated with few real users but a large number of fake user accounts controlled by attackers. In an extreme case, bad PIPs may be entirely set up by attackers. For example, it is observed that >30% of the IP addresses that issue more than 20 sign-up requests to Windows Live per day are actually controlled by attackers, with all sign-ups for malicious uses.

Classifying PIPs is a challenging task for several reasons. First, ISPs and network operators consider the size and distribution of customer populations confidential and rarely publish their network usage information. Second, some PIP addresses are dynamic, e.g., those at small coffee shops with user population sizes changing frequently. Third, good PIPs and bad PIPs can locate next to each other in the IP address space. For example, attackers can buy or compromise Web hosting IPs that are right next to the IPs of legitimate services. In addition, a good PIP can temporarily be abused. Due to these challenges, not surprisingly, it is observed that commercial proxy lists offer a low precision in identifying PIPs and provide no support for distinguishing good PIPs from bad ones [1].

Data Mining-based anomaly Detection is become prevalent in essence. Network security is just network information security. In general, all technologies and theories about secrecy, integrality, usability, reality and controllable of network information are the research domain of network security. Intrusion is an action that tries to destroy that secrecy, integrality and usability of network information, which is unlicensed and exceed authority. Intrusion Detection is a positively technology of security defend, which gets and analyses audit data of computer system and network from some network point, and to discover whether there is the action of disobeying security strategy and whether be assaulted. Intrusion Detection System is the combination of software and hardware of Intrusion Detection Data mining can be supervised & unsupervised

supervised learning is to use the available data to build one particular variable of interest in terms of rest of data.

Clustering is a division of data into groups of similar objects. Each group called cluster, consists of objects that are similar amongst them and dissimilar compared to object of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence it models data by its clusters [3].

K-mean Clustering: k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori [4]. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function knows as squared error function given by:

$$J(V)=\sum_{i=1}^{c}\sum_{j=1}^{ci}(||xi-vi||)^2$$

Where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

Algorithmic Steps for K-Means Clustering:

Let $X = \{x_1,x_2,x_3,\ldots\ldots,x_n\}$ be the set of data points and $V = \{v_1,v_2,\ldots\ldots,v_c\}$ be the set of centers.

1) Randomly select 'c' cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

$$vi = \left(\frac{1}{ci}\right)\sum_{j=1}^{ci} xi$$

Where, 'c_i' represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

Advantages:

1) Fast, robust and easier to understand.

2) Relatively efficient: O (nkd), where n is # objects, k is # clusters, d is # dimension of each object, and t is number of iterations. Normally, k, t, d << n.

3) Gives best result when data set are distinct or well separated from each other.

Disadvantages:

1) The learning algorithm requires apriori specification of the number of cluster centers.

2) The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.

3) The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results (data represented in form of Cartesian co-ordinates and polar co-ordinates will give different results).

4) Euclidean distance measures can unequally weight underlying factors.

5) The learning algorithm provides the local optima of the squared error function.

6) Randomly choosing of the cluster center cannot lead us to the fruitful result.

7) Applicable only when mean is defined i.e. fails for categorical data.

8) Unable to handle noisy data and outliers.

9) Algorithm fails for non-linear data set.

ID3 Algorithm: The ID3 algorithm (Inducing Decision Trees) was originally introduced by Quinlan in [5] and is described below in Algorithm. Here they briefly recall the steps involved in the algorithm. For a thorough discussion of the algorithm we refer the interested reader to [6].

Require: R, a set of attributes.

Require: C, the class attribute.

Require: S, data set of tuples.

1: if R is empty then

2: Return the leaf having the most frequent value in data set S.

3: else if all tuples in S have the same class value then

4: Return a leaf with that specific class value.

5: else

6: Determine attribute A with the highest information gain in S.

7: Partition S in m parts S(a1), ..., S(am) such that a1, ..., am are the different values of A.

8: Return a tree with root A and m branches labeled a1...am, such that branch i contains ID3(R − {A}, C, S (ai)).

9: end if

Decision Tree: Decision tree support tool that uses tree-like graph or models of decisions and their consequences [7][8], including event outcomes, resource costs, and utility, commonly used in operations research, in decision analysis help to identify a strategy most likely to reach a goal. In data mining and machine learning, decision tree is a predictive model that is mapping from observations about an item to conclusions about its target value. The machine learning technique for inducing a decision tree from data is called decision tree learning.

ID3 Decision Tree: Iterative Dichotomiser is an algorithm to generate a decision tree invented by Ross Quinlan, based on Occam's razor. It prefers smaller decision trees (simpler theories) over larger ones. However, it does not always produce smallest tree, and therefore heuristic. The decision tree is used by the concept of Information Entropy [9]. The ID3 Algorithm is:

1) Take all unused attributes and count their entropy concerning test samples
2) Choose attribute for which entropy is maximum
3) Make node containing that attribute
ID3 (Examples, Target _ Attribute, Attributes)

- Create a root node for the tree
- If all examples are positive, Return the single-node tree Root, with label = +.
- If all examples are negative, Return the single-node tree Root, with label = -.
- If number of predicting attributes is empty, then
- Return the single node tree Root, with label = most common value of the target attribute in the examples.
- Otherwise Begin
  - A = The Attribute that best classifies examples.
  - Decision Tree attribute for Root = A.
  - For each possible value, vi, of A,
    - Add a new tree branch below Root, corresponding to the test A = vi.
    - Let Examples (vi), be the subset of examples that have the value vi for A
      - If Examples (vi) is empty common target value in the examples
    - Else below this new branch add the sub tree ID3 (Examples(vi), Target_ Attribute, Attributes – {A}
- End
- Return Root

2. RELATED WORK

Chi-Yao Hong et. Al. proposes PIPMiner, a fully automated method to extract and classify PIPs through analyzing service logs. Our methods combine machine learning and time series analysis to distinguish good PIPs from abused ones with over 99:6% accuracy. When applying the derived PIP list to several applications, we can identify millions of malicious Windows Live accounts right on the day of their sign- ups, and detect millions of malicious Hotmail accounts well before the current detection system captures them [1].

This paper presents a novel host-based combinatorial method based on k-Means clustering and ID3 decision tree learning algorithms for unsupervised classification of anomalous and normal activities in computer network ARP traffic. The k-Means clustering method is first applied to the normal training instances to partition it into k clusters using Euclidean distance similarity. An ID3 decision tree is constructed on each cluster. Anomaly scores from the k-Means clustering algorithm and decisions of the ID3 decision trees are extracted. A special algorithm is used to combine results of the two algorithms and obtain final anomaly score values. The threshold rule is applied for making decision on the test instance normality or abnormality [10].

Bart Kuijpers et. Al. considers privacy preserving decision tree induction via ID3 in the case where the training data is vertically or vertically distributed. Furthermore, we consider the same problem in the case where the data is both vertically and vertically distributed, a situation we refer to as grid partitioned data. We give an algorithm for privacy preserving ID3 over vertically partitioned data involving more than two parties. For grid partitioned data, we discuss two different evaluation methods for preserving privacy ID3, namely, first merging vertically and developing vertically or first merging vertically and next developing vertically. Next to introducing privacy preserving data mining over grid-partitioned data, the main contribution of this paper is that we show, by means of a complexity analysis that the former evaluation method is the more efficient [11].

Cemal Cagatay Bilgin et al. review the significant contributions in the literature on complex evolving networks; metrics used from degree distribution to spectral graph analysis, real world applications from biology to social sciences, problem domains from anomaly detection, dynamic graph clustering to community detection [12].

This paper is intended to study and compare different data clustering algorithms. The algorithms under investigation are: k-means algorithm, hierarchical clustering algorithm, self-organizing maps algorithm, and expectation maximization clustering algorithm. All these algorithms are compared according to the following factors: size of dataset, number of clusters, type of dataset and type of software used [13].

K. Hanumantha Rao et al. studies the best algorithm by using classifying anomalous and normal activities in a computer networks with supervised & unsupervised algorithms that have not been used before. They analyses the algorithm that have the best efficiency or the best learning and describes the proposed system of K-means&ID3 Decision Tree [9].

Chi-Yao Hong et. Al. [1] uses various studies for their work some for result generation techniques [14] [15] [16] [17]. Here also we refer another related data from various resources such as [18] [19].

3. PROBLEM DEFINITION

The algorithm accepts a series of "training clusters," a series of sets of items and clustering's over that set. The method learns a similarity measure between item pairs to cluster future sets of items in the same fashion as the training clusters. But the SVM based clustering is not very efficient for the detection of IP addresses containing huge dataset. The ability to distinguish bad or abused populated IP ad- dresses from good ones is critical to online service using classification algorithm such PIPMiner where the classified accuracy is low.

4. PROPOSED METHODOLOGY

Here we proposed solution algorithm for K-mean to classify the data set in to number of clusters.

A. Algorithm for K-mean:

1. Pick a number (K) of cluster centers (at random)
2. Assign every item to its nearest cluster center (e.g. using Euclidean distance)
3. Move each cluster center to the mean of its assigned items
4. Repeat steps 2, 3 until convergence (change in cluster assignments less than a threshold).

B. Horizontal Partition based id3 decision tree

Input Layer:

- Define P_1, P_2… P_n Parties. (Horizontally partitioned).
- Each Party contains R set of attributes A_1, A_2, …., A_R.
- C the class attributes contains c class values C_1, C_2, …., C_c.
- For party P_i where i = 1 to n do
- If R is Empty Then
- Return a leaf node with class value
- Else If all transaction in $T(P_i)$ have the same class Then
- Return a leaf node with the class value
- Else
- Calculate Expected Information classify the given sample for each party P_i individually.
- Calculate Entropy for each attribute (A_1, A_2, …., A_R) of each party P_i.
- Calculate Information Gain for each attribute (A_1, A_2,…., A_R) of each party P_i
- Calculate Total Information Gain for each attribute of all parties (TotalInformationGain( )).
- $A_{BestAttribute} \leftarrow$ MaxInformationGain( )
- Let V_1, V_2, …., V_m be the value of attributes. $A_{BestAttribute}$ partitioned P_1, P_2,…., P_n parties into m parties
- $P_1(V_1)$, $P_1(V_2)$, …., $P_1(V_m)$
- $P_2(V_1)$, $P_2(V_2)$, …., $P_2(V_m)$
- . .
- . .
- $P_n(V_1)$, $P_n(V_2)$, …., $P_n(V_m)$
- Return the Tree whose Root is labelled $A_{BestAttribute}$ and has m edges labelled V_1, V_2, …., V_m. Such that for every i the edge Vi goes to the Tree
- NPPID3(R – $A_{BestAttribute}$, C, ($P_1(V_i)$, $P_2(V_i)$, …., $P_n(V_i)$))
- End.

5. Analysis Parameter

Here we enlist parameter for result analysis on behalf of that we analyze our result.

1. Time complexity
2. Mean Absolute Error
3. Kappa Statistics
4. Classified instances
5. Unclassified instances
6. Mean Relative Error

6. Conclusion

The populated ip address detection using the combinatorial method of clustering algorithm and the vertical partition based decision tree provides an efficient way of finding the populated ip addresses which is then used to remove the traffic overhead and the congestion or any type of attack. The proposed technique used here provides better time complexity as well as the best classification of the datasets.

References

[1] Chi-Yao Hong, Fang Yu, Yinglian Xie "Populated IP Addresses - Classification and Applications", Proceedings of the 2012 ACM conference on Computer and communications security, pp. 329-340, 2012.

[2] Ahmed Metwally and Matt Paduano "Estimating the Number of Users behind IP Addresses for Combating Abusive Traffic", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 249-257. 2011.

[3] Han J. and Kamber M. "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 2001.

[4] K. Rajalakshmi, B. Thilaka, N. Rajeswari "An adaptive K-Means Clustering Algorithm and its Application to Face Recognition", Computer Science & Mathematics, Vol. 4, no. 9, pp.88- 92. 2010.

[5] Stefano Zanero and Sergio M. Savaresi. "Unsupervised learning techniques for an intrusion detection system", Proceedings of the 2004 ACM symposium on Applied computing, pp. 412-419, 2004.

[6] Wenke Lee and S. J. Stolfo. "Data Mining Approaches for Intrusion Detection" SSYM'98 Proceedings of the 7th conference on USENIX Security Symposium - Volume 7,pp. 6-6, 1998.

[7] D. Mutz, F. Valeur, G. Vigna, and C. Kruegel, "Anomalous System Call Detection," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 61-93, Feb. 2006.

[8]. M. Thottan and C. Ji, "Anomaly Detection in IP Networks," IEEE Trans. Signal Processing, vol. 51, no. 8, pp. 2191-2204, 2003.

[9] K. Hanumantha Rao, G. Srinivas, Ankam Damodhar and M. Vikas Krishna "Implementation of Anomaly Detection Technique Using Machine Learning Algorithms", International Journal of Computer Science and Telecommunications, ISSN 2047-3338, Volume 2, Issue 3, June 2011.

[10] Yasser Yasami, Saadat Pour Mozaffari "A Novel Unsupervised Classification Approach for Network Anomaly Detection by K Means Clustering and IDs3 Decision Tree Learning Methods", The Journal of Supercomputing, Volume 53 Issue 1, pp. 231 – 245, July 2010.

[11] Bart Kuijpers, Vanessa Lemmens, Bart Moelans "Privacy Preserving ID3 over Horizontally, Vertically and Grid Partitioned Data", ArXiv Computing Research Repository (CoRR). http://arxiv.org/abs/0803.1555v1

[12] Cemal Cagatay Bilgin and B¨ulent Yener Rensselaer "Dynamic Network Evolution: Models, Clustering, Anomaly Detection", Rensselaer Polytechnic Institute, Tech. Rep.,2008

[13] Osama Abu Abbas et al. "Comparisons between Data Clustering Algorithms", The International Arab Journal of Information Technology, Vol. 5, No.-3, July 2008.

[14] GML AdaBoost Matlab Toolbox. http://goo.gl/vh0R9.

[15] Networks enterprise data acquisition and IP rotation services. http://x5.net.

[16] Quova. http://www.quova.com/.

[17] ToR network status. http://torstatus.blutmagie.de/.

[18] J. D. Brutlag. Aberrant behavior detection in time series for network monitoring. In USENIX Conference on System Administration, 2000.

[19] X. Cai and J. Heidemann. Understanding block-level address usage in the visible Internet. In SIGCOMM, 2010.