Booster in High Dimensional Data Classification

Dr. V. Sharmila[1]
M. E., Ph.D., 1 Professor,
Department of Computer Science and Engineering,
K.S.R. College of Engineering,
Tiruchengode, Tamil Nadu.

S. Ruthra[2], V. Sathish[3], M. Vijay[4],
2,3,4 UG Students,
Department of Computer Science and Engineering,
K.S.R. College of Engineering,
Tiruchengode, Tamil Nadu.

ABSTRACT— Classification problems in high dimensional data with a small number of observations are becoming more common especially in microarray data. During the last two decades, lots of efficient classification models and feature selection (FS) algorithms have been proposed for higher prediction accuracies. However, the result of an FS algorithm based on the prediction accuracy will be unstable over the variations in the training set, especially in high dimensional data. This paper proposes a new evaluation measure Q-statistic that incorporates the stability of the selected feature subset in addition to the prediction accuracy. Then, we propose the Booster of an FS algorithm that boosts the value of the Q-statistic of the algorithm applied. Empirical studies based on synthetic data and 14 microarray data sets show that Booster boosts not only the value of the Q-statistic but also the prediction accuracy of the algorithm applied unless the data set is intrinsically difficult to predict with the given algorithm.

Keywords— Traceback, Internet Traffic, Local Flow Monitoring algorithm.

I. INTRODUCTION

The presence of high dimensional data is becoming more common in many practical applications such as data mining, machine learning and microarray gene expression data analysis. Typical publicly available microarray data has tens of thousands of features with small sample size and the size of the features considered in microarray data analysis is growing. The statistical classification of the data with huge number of features and small sample size (undersampled=problem) presents an intrinsic challenge. A striking result has been found that the simple and popular Fisher linear discriminant analysis can be as poor as random guessing as the number of features gets larger.

As was reported in, most of the features of high dimensional microarray data are irrelevant to the target feature and the proportion of relevant features or the percentage of up-regulated or down-regulated genes compared with appropriate normal tissues is only 2% _ 5%. Finding relevant features simplifies learning process and increases prediction accuracy. The finding, however, should be relatively robust to the variations in training data, especially in biomedical study, since domain experts will invest considerable time and efforts on this small set of selected features. Hence, the proposed selection should provide them not only with the high predictive potential but also with the high stability in the selection.

Methods used in the problems of statistical variable selection such as forward selection, backward elimination and their combination can be used for FS problems. Most of the successful FS algorithms in high dimensional problems have utilized forward selection method but not considered backward elimination method since it is impractical to implement backward elimination process with huge number of features.

A serious intrinsic problem with forward selection is, however, a flip in the decision of the initial feature may lead to a completely different feature subset and hence the stability of the selected feature set will be very low although the selection may yield very high accuracy. This is known as the stability problem in FS. The research in this area is relatively a new field, and devising an efficient method to obtain a more stable feature subset with high accuracy is a challenging area of research.

This paper proposes Q-statistic to evaluate the performance of an FS algorithm with a classifier. This is a hybrid measure of the prediction accuracy of the classifier and the stability of the selected features. Then the paper proposes Booster on the selection of feature subset from a given FS algorithm.

The basic idea of Booster is to obtain several data sets from original data set by resampling on sample space. Then FS algorithm is applied to each of these resampled data sets to obtain different feature subsets. The union of these selected subsets will be the feature subset obtained by the Booster of FS algorithm.

II. PROBLEM STATEMENT

A. EXISTING MODEL

One often used approach is to first discretize the continuous features in the preprocessing step and use mutual information (MI) to select relevant features. This is because finding relevant features based on the discretized MI is relatively simple while finding relevant features directly from a huge number of the features with continuous values using the definition of relevancy is quite a formidable task.

Several studies based on resampling technique have been done to generate different data sets for classification problem and some of the studies utilize resampling on the feature space.

The purposes of all these studies are on the prediction accuracy of classification without consideration on the stability of the selected feature subset.

1) Drawbacks

- Most of the successful FS algorithms in high dimensional problems have utilized forward selection method but not considered backward elimination method since it is impractical to implement backward elimination process with huge number of features.
- A serious intrinsic problem with forward selection is, however, a flip in the decision of the initial feature may lead to a completely different feature subset and hence the stability of the selected feature set will be very low although the selection may yield very high accuracy.
- Devising an efficient method to obtain a more stable feature subset with high accuracy is a challenging area of research.

B. PROPOSED SYSTEM

This paper proposes Q-statistic to evaluate the performance of an FS algorithm with a classifier. This is a hybrid measure of the prediction accuracy of the classifier and the stability of the selected features. Then the paper proposes Booster on the selection of feature subset from a given FS algorithm.

The basic idea of Booster is to obtain several data sets from original data set by resampling on sample space. Then FS algorithm is applied to each of these resampled data sets to obtain different feature subsets. The union of these selected subsets will be the feature subset obtained by the Booster of FS algorithm.

Youngseok Lee and Wonchul Kang et al [1] describe an Internet flow analysis method on the cloud computing platform. Specifically, a MapReduce-based flow analysis is presented scheme that could easily process tera or peta-byte flow files collected from many routers or monitoring servers. From experiments on our testbed with four Hadoop data nodes, we achieved that flow statistics computation time for large flow files could dramatically decrease when compared with a popular flow analysis tool run on a single host. In addition, we showed that the MapReduce based flow analysis program finishes successfully against a single-machine failure.

Daniela Brauckhoff, Bernhard and Anukool Lakhina et al [2] empirically evaluate the impact of sampling on anomaly detection metrics. Starting with un-sampled flow records collected during the Blaster worm outbreak, we reconstruct the underlying packet trace and simulate packet sampling at increasing rates. We then use our knowledge of the Blaster anomaly to build a baseline of normal traffic (without Blaster), against which we can measure the anomaly size at various sampling rates. This approach allows us to evaluate the impact of packet sampling on anomaly detection without being restricted to (or biased by) a particular anomaly detection method. They are finding that packet sampling does not disturb the anomaly size when measured in volume metrics such as the number of bytes and number of packets, but grossly biases the number of flows.

Karthik Kambatla, Giorgos Kollias, et al [3] describe emerging landscape of cloud-based environments with distributed data-centers hosting large data repositories, while also providing the processing resources for analytics strongly motivates need for effective parallel/distributed algorithms. The underlying socio-economic benefits of big-data analytics and the diversity of application characteristics pose significant challenges. In the rest of this article, they are highlight the scale and scope of data analytics problems. Author describes commonly used hardware platforms for executing analytics applications, and associated considerations of storage, processing, networking, and energy. The proposed system is focus on the software substrates for applications, namely virtualization technologies, runtime systems/execution environments, and programming models. They are concluding with a brief discussion of the diverse applications of data analytics, ranging from health and human welfare to computation al modeling and simulation.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin et al [4] describe a working set of data across multiple parallel operations. This includes many iterative machine learning algorithms, as well as interactive data analysis tools. They are proposing a new framework called Spark that supports these applications while retaining the scalability and fault tolerance of MapReduce. To achieve these goals, Spark introduces an abstraction called resilient distributed datasets (RDDs).

Jun Liu, Feng Liu, and Nirwan Ansari [5] describe a Hadoop-based scalable network traffic monitoring and analysis system for big traffic data. The system is designed and implemented following a multi-layer architecture with functional components including high-speed traffic monitors, traffic collectors, data store, Map-Reduce analysis programs, result presentation interfaces, and a cluster manager. To prove the viability of the proposed system, we deploy the system into the core network of a large scale second/third generation (2G/3G) cellular network. The results demonstrate that Hadoop is a promising enabler for building an efficient, effective, and cost-efficient large-scale network traffic monitoring and analysis system.

Yeonhee Lee and Youngseok Lee [6] describe a Internet traffic measurement and analysis of characterize network usage and user behaviors, but faces the problem of scalability under the explosive growth of Internet traffic and high-speed access. Scalable Internet traffic measurement and analysis is difficult because a large data set requires matching computing and storage resources. Hadoop, an open-source computing platform of MapReduce and a distributed file system, has become a popular infrastructure for massive data analytics because it facilitates scalable data processing and storage services on a distributed computing system consisting of commodity hardware. In this paper, we present a Hadoop-based traffic monitoring system that performs IP, TCP, HTTP, and NetFlow analysis of multi-terabytes of Internet traffic in a scalable manner.

Arpit Gupta, Rudiger Birkner, Marco Canini et al [7] describe a network operators must typically perform network management tasks while coping with fixed-function network monitoring capabilities, such as IPFIX and SNMP. The advent of programmable hardware makes it possible not only to customize packet formats and protocols, but also to install custom monitoring capabilities in network devices that output data in formats that are amenable to the emerging body of scalable, distributed stream processing systems.

Vern Paxson [8] describe a stand-alone system for detecting network intruders in real-time by passively monitoring a network link over which the intruder's traffic transits. We give an overview of the system's design, which emphasizes high-speed (FDDI-rate) monitoring, real-time notification, clear separation between mechanism and policy, and extensibility. To achieve these ends, Bro is divided into an "event engine" that reduces a kernel-filtered network traffic stream into a series of higher level events, and a "policy script interpreter" that interprets event handlers written in a specialized language used to express a site' s security policy. Event handlers can update state information, synthesize new events, record information to disk, and generate real-time notifications via sys log. Author also discuss a number of attacks that attempt to subvert passive monitoring systems and defenses against these, and give particulars of how Bro analyzes the four applications integrated into it so far: Finger, FTP, Portmapper and Telnet. The system is publicly available in source code form.

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the file "MSW_USltr_format".

III. DDoS ATTACK MODEL

In the existing system, input Data is a list of segments that have the same source and destination IP/port. The result is the number of retransmission and out-of-order. First, the existing system maps the tuples in input Stream to a key-value pair, whose key is (source IP, source port, destination IP, destination port) and value is (boolean (SYN/FIN or contains data?), sequence number, payload, next expected sequence number, time stamp).

It groups the key-value pairs according to the key and obtains the lists of segments that share the same source and destination IP addresses/ports. Then we can calculate the number of retransmission and out-of-order segments for each list. Here, the packet types are not taken for loss rate analysis and study. In addition for alerting the applications is not possible. There is no concept with the feature to alert the senders. So, this paper identifies that, and helps for analyzing loss rates through end to end measurements in an efficient manner.

The proposed system is required to analyze the loss rate and change queue priority. Hence a system with efficient algorithm is required to minimize the loss rate by normal nodes. An effective and efficient IP traceback scheme against DDoS attacks based on entropy variations. It is a fundamentally different traceback mechanism from the currently adopted packet marking strategies.

Many of the available work on IP traceback depend on packet marking, either probabilistic packet marking or deterministic packet marking. Because of the vulnerability of the Internet, the packet marking mechanism suffers a number of serious drawbacks: lack of scalability; vulnerability to packet pollution from hackers and extraordinary challenge on storage space at victims or intermediate routers.

The proposed system keeps log the packet queues and drop details. The continuous packet drops are easily notified and alerting procedure is invoked to reduce the loss rate. The new approach helps in efficient packet forwarding in the router. The new system uses maximum throughput scheduling algorithm so as to serve high speed as well as normal TCP packets to flow efficiently.

On the other hand, the proposed method can work independently as an additional module on routers for monitoring and recording flow information, and communicating with its upstream and downstream routers when the pushback procedure is carried out.

IV. METHODOLOGY

A. Retransmission and Out-Of-Order Statistics

This section calculates the retransmission and out-of-order number. First, we map the tuples in input Stream to a key-value pair, whose key is (source IP, source port, destination IP, destination port) and value is (Boolean (SYN/FIN or contains data?), sequence number, payload, next expected sequence number, time stamp). It groups the key-value pairs according to the key and obtains the lists of segments that share the same source and destination IP addresses/ports. Then it calculates the number of retransmission and out-of-order segments for each list. It only counts retransmission and out-of-order of segments or for segments carrying data.

B. RTT Calculation

According to the relationship of the TCP and ACK pair, if we use the tuple of (source IP address, source port, destination IP address, destination port, next expected sequence number) as a key for the TCP segment and (destination IP address, destination port, source IP address, source port, acknowledgment number) as a key for the ACK segment, then a TCP segment and its corresponding ACK segment should share the same key. Therefore, we can group all TCP segments and their corresponding ACK segment. Then it generates a set of key-value pairs for each sender and receiver IP/port pair, whose key is (sender IP, receiver IP), value is (RTT, 1). Then, it uses reduceByKey() to sum up the values by key, obtain (total RTT, total count) for each IP pair, and use mapValues to calculate the average RTT, stored in data Stream.

C. Server Process

In this section, packet type addition, router metric information such as packet type, incoming bit rate, max packet time to live, packet resend times. During the incoming packets listening, the incoming packets log, packets sending out normally are displayed using list box controls. The packet arrival details are also displayed in chart control.

D. Client Application for LAN

In this section, the IP address of the running node is found out and used throughout the coding. The packets are generated and sent out so that the information is stored in a table directly from that node. A new record is ‘PacketsInFlow’ table is added during application load and packet count is updated each time the packets are sent. The record type is saved as LAN. These packets need not checked since they are filtered out inside the network.

E. Client Application for Incoming Routers

In this section, the IP address of the running node is found out and used throughout the coding. The packets are generated and sent out so that the information is stored in a. A new record is ‘PacketsInFlow’ table is added during application load and packet count is updated each time the packets are sent. The record type is saved as Router. These packets need to be checked using Entropy variation so that the identity flows may attack the one of the routers inside the network.

F. Entropy Variation

This section is a part of server (router) application. In this module, if there is no extraordinary change of network traffic in a very short time interval (e.g., at the level of seconds) for non-DDoS attack cases. It is true that the network traffic for a router may dynamically change a lot from peak to off-peak service times. However, this kind of change lasts for a relatively long time interval, e.g., at least at the level of minutes. If these changes are break down into seconds, the change of traffic is quite smooth in the context. The number of attack packets is at least an order of magnitude higher than that of normal flows. During a flooding attack, the number of attack packets increases dramatically. Only one DDoS attack is ongoing at a given time. It could be true that a number of attacks are ongoing concurrently in the Internet, the attack paths may overlap as well, but it only considers the one attack scenario to make it simple and clear. The local flow monitoring algorithm and IP trace back algorithm is implemented using this module.

V. EXPERIMENTAL RESULTS

The following Table 4.1 describes experimental result for Local Flow Monitoring algorithm (LFM-A). The table contains sources node id, Neighbor node id, packet size, speed and average of performances rate details are shown. The O (N) best case analysis (Performances rate) for existing LFM system is,

Performance (Local Monitoring) Rate= [(Packet Size/Speed) *60] /100

Table 4.2: Performances Analysis-Local Flow Monitoring

S.NO	Sources Node ID	Packet Size (Byte)	Speed (Minutes)	Performance Rate [%]
1	N1	1635	25	39.24
2	N2	593	10	35.58
3	N3	1365	21	39.00
4	N4	531	11	28.96
5	N5	658	13	30.36
6	N6	1677	29	34.70
7	N7	539	13	24.88
8	N8	1206	20	36.18
9	N9	1405	22	38.32
10	N10	649	12	32.45

The following Fig 4.1 describes experimental result for Local flow Monitoring algorithms. The figure contains sources node id, Neighbor node id, packet size, speed and average of performances rate details are shown.

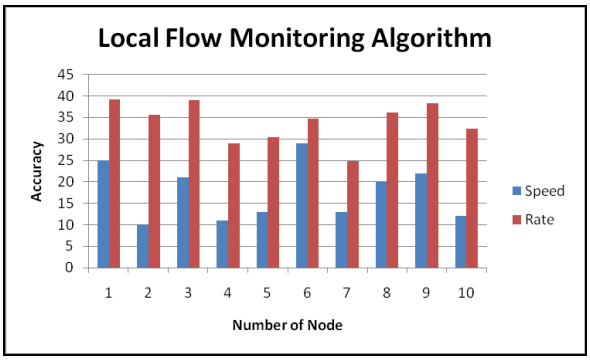

Fig 4.1 Performances Analysis-Local Flow Monitoring

The following Table 4.2 describes experimental result for IP Tracking Algorithm performances analysis. The table contains sources node id, Neighbor node id, packet size, speed and average of performances rate details are shown. The O (N) best case analysis (Performances rate) for proposed IP tracking system is,

Performance (IP Tracking Algorithm) Rate= [(Packet Size/Speed) *60] /100

Table 4.3 Performances Rate Analysis- IP Tracking Algorithm

S.NO	Sources Node ID	Packet Size (Byte)	Speed (Minutes)	Performance Rate[%]
1	N1	1635	19	51.63
2	N2	593	8	44.47
3	N3	1365	18	45.5
4	N4	531	10	31.86
5	N5	658	9	43.86
6	N6	1677	23	43.74
7	N7	539	12	26.95
8	N8	1206	17	42.56
9	N9	1405	20	42.15
10	N10	649	9	43.26

The following Figure 4.2 describes experimental result for IP Tracking Algorithm performances analysis. The figure contains sources node id, Neighbor node id, packet size, speed and average of error rate details are shown.

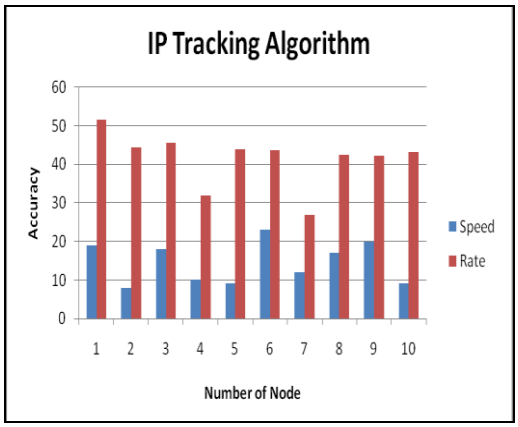

Fig 4.2 Performances Rate Analysis- IP Tracking Algorithm

VI. CONCLUSION

In this paper, it proposed an effective and efficient IP traceback scheme against DDoS attacks based on entropy variations. It is a fundamentally different traceback mechanism from the currently adopted packet marking strategies. Many of the available work on IP traceback depend on packet marking, either probabilistic packet marking or deterministic packet marking. Because of the vulnerability of the Internet, the packet marking mechanism suffers a number of serious drawbacks: lack of scalability; vulnerability to packet pollution from hackers and extraordinary challenge on

storage space at victims or intermediate routers. On the other hand, the proposed method needs no marking on packets, and therefore, avoids the inherent shortcomings of packet marking mechanisms. It employs the features that are out of the control of hackers to conduct IP traceback. It observes and store short-term information of flow entropy variations at routers. Once a DDoS attack has been identified by the victim via detection algorithms, the victim then initiates the pushback tracing procedure. The traceback algorithm first identifies its upstream routers where the attack flows came from, and then submits the traceback requests to the related upstream routers. This procedure continues until the most far away zombies are identified or when it reaches the discrimination limitation of DDoS attack flows. Extensive experiments and simulations have been conducted, and the results demonstrate that the proposed mechanism works very well in terms of effectiveness and efficiency. Compared with existing system, the proposed strategy can traceback fast in larger scale attack networks.

VII. FUTURE ENHANCEMENT

- The metric for DDoS attack flows could be further explored. The proposed method deals with the packet flooding type of attacks perfectly. However, for the attacks with small number attack packet rates, e.g., if the attack strength is less than seven times of the strength of non attack flows, then the current metric cannot discriminate it. Therefore, a metric of finer granularity is required to deal with such situations.
- Location estimation of attackers with partial information when the attack strength is less than seven times of the normal flow packet rate, the proposed method cannot succeed at the moment. However, it can detect the attack with the information that we have accumulated so far using traditional methods.
- Differentiation of the DDoS attacks and flash crowds
- In this paper, it did not consider this issue the proposed method may treat flash crowd as a DDoS attack, and therefore, resulting in false positive alarms

VIII. REFERENCES

1. Yoohwan Kim,Wing Cheong Lau,Mooi Choo Chuah and H. Jonathan Chao “PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks” IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 3, NO. 2, APRIL-JUNE 2006.
2. Michael T. Goodrich, “Probabilistic Packet Marking for Large-Scale IP Traceback” IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. X, NO. X, JANUARY 2007.
3. Shui Yu,Wanlei Zhou, and Robin Doss,” Information Theory Based Detection Against Network Behavior Mimicking DDoS Attacks” IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 4, APRIL 2008.
4. Yang Xiang, Wanlei Zhou, and MinyiGuo “Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009.
5. Akash Mittal, Prof. Ajit Kumar Shrivastava, Dr. Manish Manoria”A Review of DDOS Attack and its Countermeasures in TCP Based Networks” International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011.
6. Tao Peng, Christopher Leckie, And Kotagiri Ramamohanarao “Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems” ACM Computing Surveys, Vol. 39, No. 1, Article 3, Publication date: April 2007.
7. N. Syed Siraj Ahmed and D. P. Acharjya “Detection of Denial of Service Attack in Wireless Network using Dominance based Rough Set” (JACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 12, 2015.
8. LuKaszApiecionek, Jacek M. Czerniak, and Wojciech T. Dobrosielsk “Quality of Service Method as DDoS protection Tool” D.Fileve et al. (eds.), Intelligent Systems’ 2014, Advances in Intelligent Systems and Computing 323, Springer International Publishing Switzerland 2015.