Anomaly Detection Model using G-LDA

A PROJECT REPORT

submitted by

Bhavesh Kasliwal (10BCE1026)

Shraey Bhatia (10BCE1094)

Shubham Saini (10BCE1097)

in partial fulfillment for the award

of

B.Tech

degree in

Computer Science and Engineering

School of Computing Science and Engineering

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore ▪ Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

November - 2013

VIT
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
Vellore - 632 014, Tamil Nadu, India

School of Computing Science and Engineering

DECLARATION

We hereby declare that the project entitled **"Anomaly Detection model using G-LDA"** submitted by us to the School of Computing Science and Engineering, VIT University, Chennai Campus, Chennai – 127 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out by us under the supervision of **Prof Sumaiya Thaseen.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Signature
Bhavesh Kasliwal (10BCE1026)

Shraey Bhatia (10BCE1094)

Shubham Saini (10BCE1097)

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
www.vit.ac.in
Vellore ▪ Chennai
CHENNAI CAMPUS
Vandalur - Kelambakkam Road, Chennai - 600127

School of Computing Science and Engineering

CERTIFICATE

The project report entitled “**Anomaly Detection Model using G-LDA**” is prepared and submitted byBhavesh Kasliwal (10BCE1026) Shraey Bhatia (10BCE1094) and Shubham Saini (10BCE1097)**.** Ithas been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** in VIT University, Chennai Campus, Chennai, India.

Signature________________________

(Prof. Sumaiya Thaseen)

Examined by:

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Sumaiya Thaseen for her exemplary guidance, monitoring and constant encouragement throughout this project.

We are obliged to faculty of VIT University, for their valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of this project.

Place : Chennai

Date : 13 November, 2013

CONTENTS

Chapter	Title	Page
	Title Page	i
	Declaration	ii
	Certificate	iii
	Acknowledgement	iv
	Table of contents	v
	List of Tables	vi
	List of Figures	vii
	List of Abbreviation	viii
	Abstract	ix
1	Introduction	1
	1.1 General	1
	1.2 Motivation	1
	1.3 Problem Statement	1
	1.4 Related Work	1
	1.5 Report Organization	2
	1.6 System Information	2
2	Overview of the Proposed Work	3
	2.1 Attribute Selection	3
	2.2 Set generation	5
	2.3 Genetic Algorithm	6
	2.4 Determining the type of packet	6
3	Implementation	7
4	Results and Discussion	10
5	Conclusion and Future Enhancements	12
	References	13

LIST OF FIGURES

Fig 3.1 : G-LDA Process

Fig 3.2: The GA Process

LIST OF ABBREVIATIONS

Abbreviation	**Expansion**
LDA	Latent Dirichlet Allocation
GA	Genetic Algorithm
SVM	Support Vector Machine
JDK	Java Development Kit

ABSTRACT

Intrusion detection is one of the important challenges of network security associated today. We present a novel technique called G-LDA to identify the anomalies in network traffic. We propose a hybrid technique integrating Latent Dirichlet Allocation and genetic algorithm namely the G-LDA process. Our implementation and performance analysis was done on KDDCUP'99 dataset. Furthermore, feature selection plays an important role in identifying the subset of attributes for determining the anomaly packets. Our analysis shows the hybrid implementation of G-LDA and in depth performance analysis of the proposed model.

LDA is a way of automatically discovering topics within a documents collection. Each word in the documents collection will belong to a particular topic. The technique can be applied to a spam email training data. The result will be a set of words having a high probability of occurrence within a spam email.

CHAPTER 1

INTRODUCTION

1.1 General

An **intrusion detection system** (**IDS**) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

IDPSs typically record information related to observed events, notify security administrators of important observed events and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content. They generally comprise of statistical anomaly based IDS and signature based IDS. **Network Intrusion Detection System** (**NIDS**) is an intrusion detection system that attempts to discover unauthorized access to a computer network by analyzing traffic on the network for signs on malicious activity.

1.2 Motivation

The motivation to go for this problem was to come out with a novel concept to be able to identify network intrusions, more specifically anomaly detection. By using LDA and genetic analysis we came out with G-LDA to help us in increasing the performance of spam identification over the conventional method. The work done Benjamin Newton and Crammer on LDA for anomaly detection gave us a starting point. Moreover, We lie’s work on genetic algorithms gave us further insight into it.

1.3 Problem Description

Detection of anomalies using Latent Dirichlet Allocation and Genetic Evolution techniques. Our aim is to use probabilistic approach followed by Genetic Analysis to be able to identify anomalies. LDA is based on probability and helps us in giving a set of anomaly packets. Genetic Evolution is applied to previous set to get prepare fields of network packet with improved fitness values and thus helps us in improved detection of anomalies

1.4 Related Work

The literature indicates that authors have employed Latent Dirichlet Allocation (LDA) and Genetic algorithms individually for network modeling and intrusion identification. Some of them are as follows.LDA for traffic analysis was initially employed by Cramer et al .Benjamin D. Newton et al applied LDA for anomaly detection on network traces at University of North Carolina at Chapel Hill

(UNC).The authors used LDA on packet counts,user sessions, documents and port numbers.Moreover, We Lie et al used genetic algorithms for intrusion detection. Zhang et al employed LDA to provide an accurate analysis of whether the network traffic model is of actual traffic type. Using the linear discriminant arithmetic, data sets generated by a network simulator were analyzed for identification of complex network traffic. Gomathy et al proposed feature selection by employing genetic algorithm for better accuracy and efficiency.

1.5 Report Organization:

The report includes description of proposed work followed by system requirements. Implementation, proof of concept is explained and is supported by results in the next section. Finally, it ends with conclusion and future enhancements of our work.

1.6 System information

The system uses MySQL as database. Java is used as a programming language. KDDCUP’99 is used as our set for training and to conduct tests and implementation

CHAPTER 2

OVERVIEW OF THE PROPOSED WORK

2.1 Attribute Selection

There are certain attributes in a network packet that play a major role in indentifying the nature of the packet such as normal or anomaly. The mean and mode values of each numerical attribute for both anomaly and normal packets are calculated as this measure is used to identify the best feature subset. The attributes having different mode values for the anomaly and normal packets with their mean close to their mode value were chosen for anomaly detection purpose. Tables and show the mode and mean values for the attributes and table III show the best feature subset identified from tables 1 and2.

Attribute	Anomaly Mode	Normal Mode
duration'	0	0
'src_bytes'	0	0
'dst_bytes'	0	0
'land'	0	0
'wrong_fragment'	0	0
'urgent'	0	0
'hot'	0	0
'num_failed_logins'	0	0
'logged_in'	0	1
'num_compromised'	0	0
'root_shell'	0	0
'su_attempted'	0	0
'num_root'	0	0
'num_file_creations'	0	0
'num_shells'	0	0
'num_access_files'	0	0
'num_outbound_cmds'	0	0
'is_host_login'	0	0
'is_guest_login'	0	0
'count'	1	1
'srv_count'	1	1
'serror_rate'	1	0
'srv_serror_rate'	1	0
'rerror_rate'	0	0
'srv_rerror_rate'	0	0
'same_srv_rate'	1	1
'diff_srv_rate'	0.06	0
'srv_diff_host_rate'	0	0

'dst_host_count'	255	255
'dst_host_srv_count'	1	255
'dst_host_same_srv_rate'	1	1
'dst_host_diff_srv_rate'	0.07	0
'dst_host_same_src_port_rate'	0	0
'dst_host_srv_diff_host_rate'	0	0
'dst_host_serror_rate'	1	0
'dst_host_srv_serror_rate'	1	0
'dst_host_rerror_rate'	0	0

Attribute	**Anomaly Mean**	**Normal Mean**
duration'	423.32069	168.5873959
'src_bytes'	82820.141	13133.27933
'dst_bytes'	37524.482	4329.685223
'land'	0.000307	0.000103945
'wrong_fragment'	0.0487464	0
'urgent'	6.82E-05	0.000148494
'hot'	0.1742623	0.230655005
'num_failed_logins'	0.0010404	0.00138099
'logged_in'	0.0340269	0.710645501
'num_compromised'	0.0175678	0.507075717
'root_shell'	0.0005458	0.002034361
'su_attempted'	1.71E-05	0.002049211
'num_root'	0.0027119	0.562924135
'num_file_creations'	0.0016374	0.02227403
'num_shells'	0.0001876	0.000608823
'num_access_files'	0.0001876	0.007498923
'num_outbound_cmds'	0	0

'is_host_login'	0	1.48E-05
'is_guest_login'	0.0053556	0.012963485
'count'	154.84999	22.51794544
'srv_count'	27.797885	27.68565404
'serror_rate'	0.5958079	0.013440892
'srv_serror_rate'	0.5930718	0.012083364
'rerror_rate'	0.20698	0.044195982
'srv_rerror_rate'	0.2091141	0.044629137
'same_srv_rate'	0.306659	0.969360141
'diff_srv_rate'	0.1024095	0.028787847
'srv_diff_host_rate'	0.064079	0.126263309
'dst_host_count'	222.02526	147.4319231
'dst_host_srv_count'	29.929081	190.285761
'dst_host_same_srv_rate'	0.1874174	0.811875028
'dst_host_diff_srv_rate'	0.1321313	0.040133941
'dst_host_same_src_port_rate'	0.1789928	0.121725792
'dst_host_srv_diff_host_rate'	0.0400621	0.025995723
'dst_host_serror_rate'	0.5951772	0.01393003
'dst_host_srv_serror_rate'	0.5913292	0.006116449
'dst_host_rerror_rate'	0.2018103	0.046589252

S.no	Feature
1	logged_in
2	Serror_rate
3	srv_serror_rate
4	Same_srv_rate
5	diff_srv_rate
6	dst_host_serror_rate
7	dst_host_srv_serror_rate

2.2 Set Generation

In this paper, the KDDCUP ’99 dataset is used which is based on the 1998 DARPA. The KDDCUP’99 is the most widely used data set for research in KDD and data mining. We employ this data set for construction and evaluation of our anomaly based model. After converting into a required format, KDD data set is supplied to LDA using JGibbLDA [13] package. JGibbLDA is a Java implementation of LDA using Gibbs Sampling technique. A total of 200 sets with 10 words each were generated from the anomaly packets. Each set contained a single packet type as majority. Although there were not 200 different packet types in the data set, it was found that distributing the packets over a large number of sets gave the best results. The similar process was applied to the normal packets also to obtain unique sets.

2.3 Genetic Algorithm

The attribute subset is retrieved from each of the 2000 generated packets separately and genetic evolution is applied on every attribute.

- Loop for n generations
- Find Fitness Function for each number
- Perform breeding using roulette wheel selection
- Filtering to produce a new generation
- Calculate the fitness value.

  $$\text{Fitness} = \text{Number of packets}*(1+ \frac{N-N_w}{N})$$

  Where

  N = Threshold value for restricting the number of generations (Taken it as 2)

  N_w = No. of values in the term

2.4 Determining the type of incoming packet

For determining if an incoming packet is anomaly or not, the following procedure is performed:

```
• For each selected attribute value $F_i$ in incoming packet
      If $F_i \in V_i$
              $S_i$ = (A* Frequency of $F_i$ in Anomaly) – (Frequency of $F_i$ in Normal)
      Else
              $S_i = 0$
• $C = \sum S_i$
• If $C > 0$
      o Then Anomaly
• Else Normal
```

Here A is the additional weight that is multiplied to the anomaly frequency.

CHAPTER 3

IMPLEMENTATION

The system was developed using the JDK 7.0 built over the Eclipse IDE. The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. Java is the closest thing to a lingua franca (the idiom means "common language") we have in the industry today. Just about every has either used Java at some point, or (more commonly) is actively using it now. That sort of ubiquity can be extremely attractive to some companies, particularly those reliant on consultants. Java is an excellent language for developing cross-platform desktop applications.

Reasons for choosing Java over other programming languages:

- price - it's free
- performance - really fast these days thanks to the HotSpot JIT compiler
- effectiveness - lots of power with rigorous features like type-safe, sand-boxed, etc.
- OOP capability
- very good, well-thought out exception handliing; C++ exceptions are the opposite!
- portability - it runs on almost everything
- tool availability - awesome IDEs like Eclipse & NetBeans are free, as are web servers like Tomcat and application servers (JBoss, Glassfish, Geronimo, etc.)
- flexibility - does graphics, desktop GUIs, web user interfaces - all kinds of things in all kinds of runtime environments
- aptness - many enterprise apps today have to support HTML, SQL, and XML - Java has good support for all of them built in and you can get third-party libraries for free that make this even easier/better
- well-supported - Sun keeps adding improvements and fixing thing going one or two versions back
- forward compatibility - unlike something like VB which undergoes wrenching change in its syntax every couple versions or so, Java's syntax and semantics seem about 99.9% upward compatible from version to version

JGibbLDA

JGibbLDA is a Java implementation of Latent Dirichlet Allocation (LDA) using Gibbs Sampling technique for parameter estimation and inference. The input and output for JGibbLDA are the same format as GibbLDA++. Because the parameter inference process require less computational time than parameter estimation, JGibbLDA focus on infering hidden/latent topic structures of unseen data upon the model estimated using GibbLDA++ . It also provides a convenient API to get topic structures for an array of input strings.

JGibbLDA is useful for the following potential application areas:

- Information Retrieval (analyzing semantic/latent topic/concept structures of large text collection for a more intelligent information search.
- Document Classification/Clustering, Document Summarization, and Text/Web Data Mining community in general.
- Collaborative Filtering
- Content-based Image Clustering, Object Recognition, and other applications of Computer Vision in general.
- Other potential applications in biological data.

THE G-LDA PROCESS

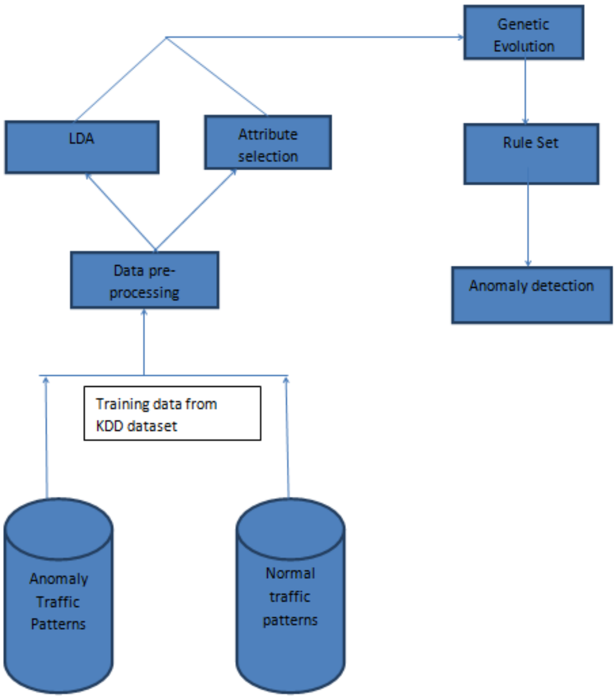

Fig 3.1 – The G-LDA Process

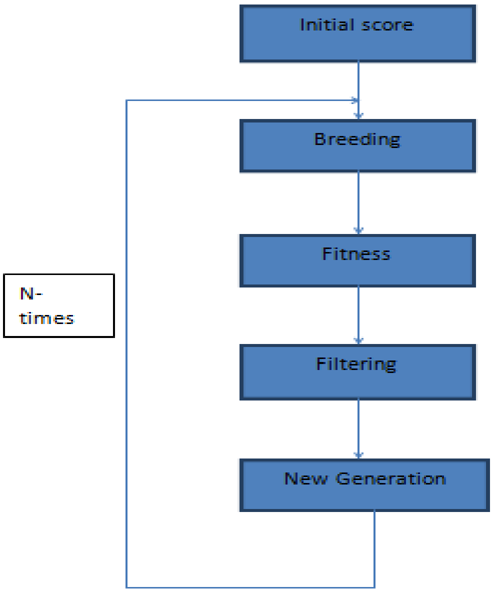

Fig 3.2 –GA component

CHAPTER 4

RESULTS AND DISCUSSION

The experiment was done on the Intel i5 processor with 4GB memory in Windows environment. All the programming was done using Java SDK 7. The output from the Set Generation stage were imported into a MySQL database, which were then processed via Java using the JDBC interface. For testing the efficiency of the rule set, 50000 Anomaly and 50000 Normal class packets from the KDDCUP'99 set were used

Table shows the number of sample instances taken for training and test data set. We measure the performance of our model using the following metrics. These values are true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

i) Accuracy (Acc): (TN+TP)/(TN+TP+FN+FP);
Proportion of the total number of predictions that were correct.

ii) Precision Rate (PR): TP/(TP+FP)
Proportion of the predictive positive cases that were correct.

iii) Detection Rate (DR) : TP/(TP+FN)
Number of intrusion samples detected by the model (True Positive) divided by the total number of intrusion samples present in the test set.

iv) False Positive Rate (FPR): FP/(TN+FP)
Number of samples misclassified as anomalies.

Tables 4 and 5 summarize the performance metrics for various additional weights assigned to the anomaly frequency to determine the optimized accuracy for the proposed model. We conclude that when the weight is 1.75, we obtain 0.885 accuracy with a false positive rate of 0.06. Hence this weight measure can be fixed for identifying whether an incoming packet is of normal or abnormal type.

	Training Data	Test Data
Anomaly	2000	50000
Normal	2000	50000

Weight	TP	TN	FP	FN
1	60	99.75	0.25	40
1.5	64	99	1	36
1.7	80	95	5	20
1.75	83	94	6	17
1.8	86	91	9	14
1.85	88	78	22	12
2	96.5	70	30	3.5

Weight	DR	PR	Acc	FPR	F-score
1	0.6	0.995851	0.79875	0.0025	0.74883
1.5	0.64	0.984615	0.815	0.01	0.775758
1.7	0.8	0.941176	0.875	0.05	0.864865
1.75	0.83	0.932584	0.885	0.06	0.878307
1.8	0.86	0.905263	0.885	0.09	0.882051
1.85	0.88	0.8	0.83	0.22	0.838095
2	0.965	0.762846	0.8325	0.3	0.852097

CHAPTER 5

CONCLUSIONS AND FUTURE ENHANCEMENTS

We propose a novel technique for detecting anomalous network traffic using G-LDA. Latent Dirichlet Allocation has been proved useful for generating a model of the network data to identify anomalies in network traffic. Only a subset of all attributes of a network packet was chosen for anomaly detection purpose. As network attacks become more sophisticated and unpredictable continuously, performing Genetic Evolution techniques to the modeled network data allows us to detect previously unknown anomalous network data.

We evaluated the performance of our proposed system. The G-LDA process for detecting anomalous network traffic showed an accuracy of 88.5%, with a 6% false positive Rate.

It may be noted that the process was applied on a generic anomaly data. Applying the same over a specific anomaly type may lead to an increased accuracy, and is intended as future work.

REFERENCES

[1] Valeur, Fredrik, and Giovanni Vigna. Intrusion detection and correlation: challenges and solutions. Vol. 14. Springer, 2005.

[2] Kim, Dong Seong, and Jong Sou Park. "Network-based intrusion detection with support vector machines." Information Networking. Springer Berlin Heidelberg, 2003.

[3] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research,Volume 3, pp.993-1022,2003.

[4] Cramer, Christopher, and Lawrence Carin. "Bayesian topic models for describing computer network behaviors." Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011.

[5] Newton, Benjamin D. "Anomaly Detection in Network Traffic Traces Using Latent Dirichlet Allocation."

[6] Li, Wei. "Using genetic algorithm for network intrusion detection." Proceedings of the United States Department of Energy Cyber Security Group,pp1-8,2004.

[7] Bing-Yi Zhang,Ya-Min Sun,Yu-Lan,Bian,Hong Ke Zhang,"Linear Discriminant Analysis in network traffic modeling", International Journal of Communication Systems",Volume 19,Issue 1,pp.53-65,2006.

[8] A.Gomathy and B.Lakshmi,"Network intrusion detection using Genetic algorithm and Neural Network", Communications in Computer and Information Science,Volume 198,pp.399-408,2011.

[9] Siva S,Sivatha Sindhu,S.Geetha,A.Kannan,"Decision tree based light weight intrusion detection using a wrapper approach",Expert Systems with applications,Volume 39,pp.129-141,2012.

[10] B.Kavitha,S.Karthikeyan,P.Sheeba Maybell,"An ensemble design of intrusion detection system for handling uncertainity using neutrosophic logicclassifier",Knowledge based systems,Volume 28,pp.88-96,2012.

[11] Saini, Shubham, Bhavesh Kasliwal, and Shraey Bhatia. "Spam Detection using G-LDA." International Journal of Advanced Research in Computer Science and Software Engineering,Volume 3,Issue 10,pp.406-409,2013.

[12] Cup, K. D. D. "Available on: http://kdd. ics. uci. edu/databases/kddcup 99/kddcup99. html.",2007.