Journal of Universal Computer Science, vol. 22, no. 4 (2016), 589-604
submitted: 1/10/15, accepted: 30/3/16, appeared: 1/4/16 © J.UCS

An Approach for Intrusion Detection Using Novel Gaussian Based Kernel Function

Gunupudi Rajesh Kumar
(Dept of Information Technology, VNRVJIET, Hyderabad, India
gunupudirajesh@gmail.com)

Nimmala Mangathayaru
(Dept of Information Technology, VNRVJIET, Hyderabad, India
mangathayaru_n@vnrvjiet.in)

Gugulothu Narsimha
(Jawaharlal Nehru Technological University, Hyderabad, India
narsimha06@gmail.com)

Abstract: Software Security and Intrusion Detection need to be dealt at three levels Network, Host level and Application level. In this paper the major objective is to design and analyze the suitability of Gaussian similarity measure for intrusion detection. The objective is to use this as a distance measure to find the distance between any two data samples of training set such as DARPA Data Set, KDD Data Set. This major objective is to use this measure as a distance metric when applying k-means algorithm. The novelty of this approach is making use of the proposed distance function as part of k-means algorithm so as to obtain disjoint clusters. This is followed by a case study, which demonstrates the process of Intrusion Detection. The proposed similarity has fixed upper and lower bounds.

Key Words: Intrusion Detection, Similarity Function, k-Means, Gaussian, Text Processing, Software Vulnerabilities
Category: C.2.0, K.6.5, I.5.4

1 Introduction

The Intrusion Detection is the process of acquiring or an unauthorized attempt, to acquire the rights over computing resources or information resources. Nowadays Intrusion Detection is becoming an alarming problem. Research in this area is started many years back and there were significant improvements in the intrusion detection process. The attacks and threats are also changing their orientation while this aggression.

Several Intrusion Detection Systems are in use which are working on different approaches such as Signature based [Hubballi, 2014], anomaly based, SVM Based [Joachims, 1999], Text Processing, Genetic algorithm based, Fuzzy Logic based [Abadeh, 2011] and Association Rule based approaches. Apart from all these approaches if the intrusion detection mechanisms are to be devised in two broad categories, they are signature based and anomaly based techniques [Treinen, 2006] [Stolfo, 1999].

Signature based Intrusion Detection System (IDS) is generally works based on analyzing the packets, packet sequences, traffic analysis. The IDS will search into the packet for some sequence or pattern which we call as a signature known to be malicious [Hyun, 2003]. The Signature based detection approach gives fruitful results only for the known attacks. The advantage of these approaches is the identifying a signature for a threat and loading its pattern into the database is quite simple [Meng, 2013]. Once these signatures were loaded into the database, the IDS will check each packet and compare whether the signature pattern is present in the packet or not in the packet or bit sequence. The Signature matching engines do have their own disadvantages as they detect only known attacks. This approach is not suitable for those attacks which were not present in the Signature database. The rate of getting false alarms in this case are huge in size. The reason for getting false alarms very frequently, is that generally signatures consists of regular expressions, string patterns. While the signature based detection shows an excellent performance in the case of threats consisting of fixed behavior patterns. Whereas, it is almost impossible to detect unknown and threats that do frequently changes behavior as in attacks generated through intelligent softwares such as worms, trojans, etc., as they have self-modifying behavioral characteristics. A Signature based IDS introducing the arms race between the IDS Signature developers and attackers. The performance of the signature based IDS is greatly influenced by the size of the signature data base as it has accelerated growth in volume. Even Small variation in the signature, causing a new entry in a signature database [Modi, 2013] [Govindarajan, 2011].

The anomaly based Intrusion Detection System basically works on the principle of creating boundaries which specifies accepted behavior and unaccepted behavior. Any incoming event or outgoing event which falls in the range of unaccepted behavior in an anomaly detection engine declares it as a threat. The important point while designing the anomaly detection engine is that the engine must be given power to get into deep of each of the protocols that need to be monitored by the engine. Of course this is a very expensive job as dissection of the protocols in the initial stage is complex. The biggest challenge of the anomaly based IDS, is to understand, design, test and implement the rules for each protocol. On the other hand, if the rules are formed, the anomaly based IDS performs threat a detection job can be scaled more quickly and easily than the signature based IDS. The major pitfall of anomaly based IDS is that any anomaly within the range of the normal usage patterns is undetected. However, the anomaly based IDS is far better than the signature based IDS as any new threat not having the signature will be detected as its behaviour is out of the normal behaviour pattern [Sharma, 2007].

In this paper, we use a novel similarity measure to form the normal behaviour over the system calls caused by the processes. A case study is also discussed, to describe how the train data is useful in detecting the intrusions. The dimensionality reduction process is used to simplify the pre-processed data in making decisions. Different techniques like clustering, nearest neighbour concepts are used to identify each process with unique value which is called as similarity, thus enabling simpler detection process.

In Intrusion detection at application level, if the mechanisms such as semantic data validations prevent the attacks even if the attacks bypassed at the network or host level [Aljawarneh, 2016] [Pistoia, 2015] [Hsu, 2011]. In [Aljawarneh, 2010] the researchers explore threats and address challenges posed by polymorphic worms to internet infrastructure security.

2 Related Study

Software development is complex. On adding, problem complexity, design complexity, program complexity, the difficulty level in the software development, errors, bugs, failures, faults may arise in any stage of the development process. The consequences of these errors, bugs, failures or faults may lead to the software vulnerabilities [Krsul, 1998]. The Software Security, generally be provided in two levels, the first one is at application level and the other one is to provide at the network level. The security at network level with the help of firewall is good, but it cannot provide the data integrity at the application level. They are useless in case any malware, malicious code is already present in the node behind the firewall, which cannot be detected by the firewall. With the increase in the usage of mobile devices in the network, this security threat is becoming more severe problem [Aljawarneh, 2016] [Aljawarneh, 2010].

In web applications, the SQL injection is one of the major vulnerability which need to be addressed carefully. Poor data processing techniques cause the SQL injection attacks. Almost many of the software problems are based on the instances of general patterns like buffer overflows, string pattern vulnerabilities [Nguyen, 2010].

It is almost difficult to safeguard computer networks against any possible attack. Without proper selection and utilization of tool support, the security of computer network is very much risky and labour intensive and error prone, as the presence of complexity, size, and dynamic changes that present in the network configuration. Software vulnerabilities are very common and many times the patches, solutions are not readily be provided by any of the tool. Importantly, these security concerns are interdependent across the entire network.

More frequently the attackers can attack only on vulnerable machines and make use of them as stepping stones to penetrate further into the network and results in compromise of critical systems. The solutions available in now a days are point oriented solutions, giving a few clues for defence of strategic network. It is almost impossible or very difficult to combine the results from multiple tools and data sources for the protection mechanism. It is a challenging task for even experienced analysts to identify threats like multi-step attack risks, and to understand which vulnerabilities really are acceptable risks.

The analysis is almost challenging for networks that are spreaded over the different places with wide varieties of technologies and protocols and the nature of dynamically changing. By understanding the solutions of vulnerabilities through the computer networks, we obviously can reduce the impact of attacks. It is almost impossible to relay on either single tool or technology or approach in identifying the vulnerabilities. In contrast, the traditional network vulnerability detection tools simply scan individual machines over a network and report only few possible security problems. These tools only give a little guidance how the attackers are going to exploit with the help of different combinations of vulnerabilities among multiple hosts in order to advance an attack over a network [Krsul, 1998]. Different functions performed by Intrusion Detection are as follows:

1. Analysing and Continuous monitoring of system and user activity
2. Auditing job of different vulnerabilities and system configurations
3. Evaluating and Judgment of the integrity of critical system and data files
4. Identification and Statistical analysis of abnormal activity patterns
5. Operating system audit trace management, with identification of user activity reflecting violation of policy

Benefits of products engaged in intrusion detection and assessment of vulnerability, include the following:

1. Improving the integrity of information technology security infrastructure.
2. Improving system monitoring and user activity tracing from entry point to exit point or impact.
3. Identification and reporting of alterations made to data files and error spotting of system configurations and making corrections if needed.
4. Identification of specific attack types and raising an alert to appropriate staff in order to provide defense mechanism.
5. Keeping security management staff updated on latest corrections to settings and programs.

6. Providing user friendly environment to non-expert personnel to contribute to security and providing guidelines in security policies.

3 Various Knowledge Discovery Based Approaches for Intrusion Detection

Most of the significant works carried for finding intrusion detection may be classified into the following classes

1. Intrusion detection based on Machine Learning
2. Unsupervised Learning based Intrusion Detection
3. Intrusion detection based on Supervised Learning

3.1 Intrusion detection based on Machine Learning

Machine learning is a self-learning approach which requires a formal system which can update itself continuously each time the new data is generated and added to the system. In essence, it must be an autonomous system which can address the continuous changes coined out and integrate the knowledge database. This process requires ability to learn from experience, analytical capability, self-learning capability, ability to handle dynamic changes to get self-updated. In essence, the major task in the machine learning algorithms is to design, analyze, develop, and implement various algorithms and methodologies which guide the machines (computer systems) to gain self-learning capability. Machine learning may be classified into supervised and unsupervised learning techniques [Lin, 2015].

3.2 Intrusion Detection Based on Supervised

In this approach for intrusion detection, we must know the class label to build the knowledge database or knowledge rules. This is because of this reason; we call it as supervised learning technique or classification. Given a dataset, we split the dataset into training and testing sets, and build the knowledge using the training set. Then we use samples from the testing set to test the class label of the test-case chosen from the testing test. In short, the task of supervised learning is to build a classifier which can effectively approximate the mapping between input and output samples of training. Once we build a classifier, it followed by measuring the classification accuracy. Classification requires choosing an appropriate function which can estimate the class label. This is followed by measurement of the classification accuracy. The most popular classifiers include the Decision tree based Classifier, ANN based classifier, kNN Classifier, SVM

Classifier. The simplest non-parameter classifier is kNN-classifier which is used to estimate the class label of the test input by assigning the label of the nearest neighbor.

3.3 Intrusion Detection Based on Unsupervised Learning Technique

In the intrusion detection based on supervised learning technique, we do not have any knowledge on the class labels of the input dataset. In such a situation, we aim to choose the classifier based unsupervised learning. This process is also called as clustering process. In unsupervised learning based technique the objective is to obtain a disjoint set of groups consisting of similar input objects. These groups may be used to perform decision making, to predict future inputs. The k-Means clustering method is the most popular among the various clustering algorithms where k indicates the number of clusters to be formed from the input dataset. The k-Means algorithm requires specifying the number of clusters to be formed well ahead. In [Lin, 2015], the authors make use of this property to decide the number of clusters in their approach for intrusion detection [Barbar, 2001] [Manganaris, 1999].

4 Proposed Approach

The consensus based computing approach has been applied in various application areas which aims at using more than one algorithm or procedure, distance measures to ad- dress the respective problems. Since the chosen dataset has already defined the number of classes, and the intrusion detection is also a classification problem, we may choose to cluster the chosen dataset into a number of clusters equal to the number of class labels. In this paper, the objective is to use the k-Means clustering method to cluster the chosen dataset into a number of clusters equal to the number of class labels. We may directly cluster the training set or alternatively choose perform feature selection followed by dimensionality reduction and then apply k-Means clustering over this reduced dimensionality [Kumar1, 2015] [Kumar2, 2015].

The consensus based computing approach has been applied in various application areas which aims at using more than one algorithm or procedure, distance measures to ad- dress the respective problems. Since the chosen dataset has already defined the number of classes, and the intrusion detection is also a classification problem, we may choose to cluster the chosen dataset into a number of clusters equal to the number of class labels. In this paper, the objective is to use the k-Means clustering method to cluster the chosen dataset into a number of clusters equal to the number of class labels. We may directly cluster the training set or alternatively choose perform feature selection followed by dimensionality reduction and then apply k-Means clustering over this reduced

dimensionality [Kumar3, 2015] [Kumar4, 2016] [Kumar5, 2016]. We follow the approach in [Lin, 2015] for dimensionality reduction. However, instead of using the conventional k-means algorithm, we choose to apply the modified k-Means algorithm which uses the Gaussian based distance measure to find the similarity between data samples when forming the clusters. This is where the novelty of our approach starts with. In this approach, we reduce the dimensionality of the training set by first applying a suitable clustering to a number of clusters equal to a number of known class labels. Since the intrusion datasets have labeled attacks, we can decide the number of clusters to be obtained. The better choice is k-Means clustering algorithm as it clusters the input to the predefined number of clusters [Lee, 1998].

After, obtaining the clusters, the next step is to find the distance between each training data sample and all the cluster centres. This is the first distance value computed. In addition to this for every data sample with in a cluster, we find its nearest neighbour within that cluster by selecting the pair of minimum distance. This is the second distance value [Vapnik, 1995].

4.1 Distance Measure for k-Means

In this section, we discuss the distance measure used as part of the k-Means clustering algorithm. We use the Gaussian function as the distance measure to find the distance between any two samples of training set. This may also be used to find the distance between any two data samples in general.

4.2 Gaussian Function

We consider the Gaussian function based distance measure to find the similarity between the data samples of the intrusion dataset. We use the same distance measure and apply k-Means algorithm to cluster the data samples. For the purpose of dimensionality reduction, we use the k-Means clustering technique to obtain clusters using the proposed distance function and then, to find the distance between each training data sample and each of the cluster centroids. This is further followed by finding the nearest neighbour for every data sample within the cluster. These two distances are summed to get a new distance value. This distance value becomes singleton feature for each training data sample. Thus each data sample of the training set is mapped to a single feature value reducing the dimensionality to 1.

The Proposed distance function is defined as given in Equation. 5. We consider the Gaussian function based distance measure to find the similarity between processes of the intrusion dataset. We use the same distance measure and apply k-Means algorithm to cluster the processes. For the purpose of dimensionality reduction, we use the k-Means clustering technique to obtain clusters using the

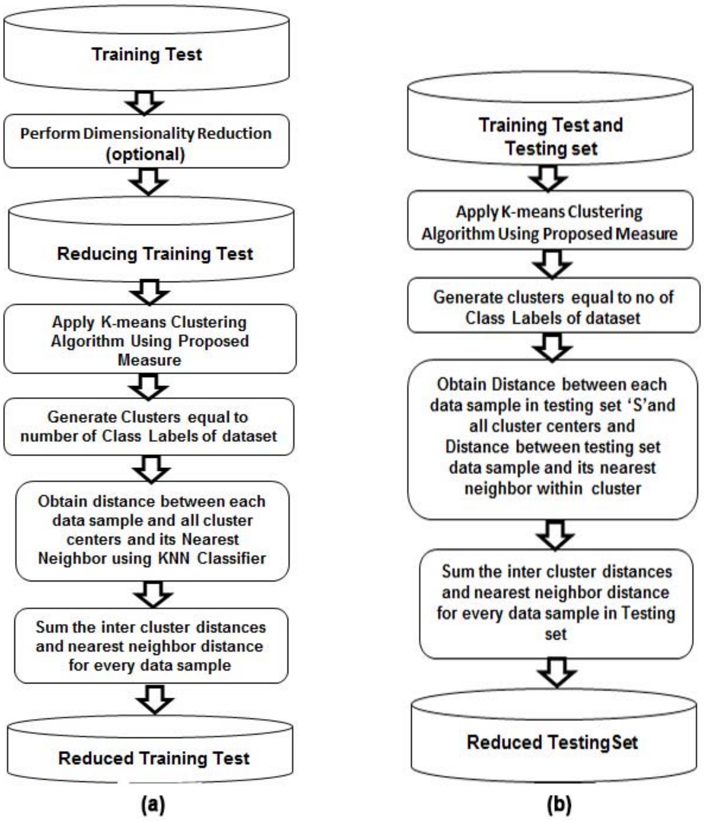
Training Test
Perform Dimensionality Reduction (optional)
Reducing Training Test
Apply K-means Clustering Algorithm Using Proposed Measure
Generate Clusters equal to number of Class Labels of dataset
Obtain distance between each data sample and all cluster centers and its Nearest Neighbor using KNN Classifier
Sum the inter cluster distances and nearest neighbor distance for every data sample
Reduced Training Test
(a)
Training Test and Testing set
Apply K-means Clustering Algorithm Using Proposed Measure
Generate clusters equal to no of Class Labels of dataset
Obtain Distance between each data sample in testing set 'S'and all cluster centers and Distance between testing set data sample and its nearest neighbor within cluster
Sum the inter cluster distances and nearest neighbor distance for every data sample in Testing set
Reduced Testing Set
(b)

Figure 1: (a) Dimensionality Reduction of Training set (b) Dimensionality Reduction of Testing Set

proposed distance function and then find the distance between each training data sample and each of the cluster centroids. This is further followed by finding the nearest neighbour for every data sample within the cluster. These two distances are summed to get a new distance value. This distance value becomes singleton feature for each training data sample. Thus each data sample of the training set is mapped to a single feature value reducing the dimensionality to 1.

$$G(x,\mu,\sigma) = \begin{cases} e^{-(\frac{x-\mu}{\sigma})^2} & ; \quad \textit{one or both system calls exist} \\ 0 & ; \quad \textit{none of the system calls exist} \end{cases} \tag{1}$$

where
x = system call being considered

μ = mean of the system call w.r.t data samples present in the cluster

σ= standard deviation of the system call considered w.r.t data samples of the training set.

The denominator of IDSIM is given in Equation.2 as shown below

$$H(x,\mu,\sigma) = \begin{cases} 1 \quad \textit{one or both system calls exist} \\ 0 \; none\ of\ the\ system\ calls\ exist \end{cases} \tag{2}$$

The average similarity is the ratio of $G(x,\mu,\sigma)$ and $H(x,\mu,\sigma)$ and is represented as given by Equation 3.

$$\frac{G(x,\mu,\sigma)}{H(x,\mu,\sigma)} \tag{3}$$

The average similarity considering the distribution of all features hence is defined as the ration of $G(x,\mu,\sigma)$ and $H(x,\mu,\sigma)$ which is reduced to Equation.4 as given below

$$F_{avg} = \frac{\sum_{i=1}^{i=n} 1 \quad \sum_{s=1}^{s=m} \quad e^{-(\frac{x_{is}-\mu_{is}}{\sigma_s})^2}}{\sum_{s=1}^{s=m} 1} \tag{4}$$

The similarity function is represented as given by

$$IDSIM = (1 + F_{avg})/2 \tag{5}$$

Where i indicates the ith data sample. S indicates the system call . IDSIM indicates the similarity function.

We may define distance value as

$$dist = 1 - IDSim \tag{6}$$

4.3 Dimensionality Reduction of Training Set for Intrusion Detection

Figure.1 (a) shows the proposed approach for reducing the dimensionality of the training set and Figure.1 (b) shows the proposed approach for reducing the dimensionality of the testing set using the proposed measure with k-Means clustering technique. So, we have both the testing and training sets with each data sample transformed to a singleton feature value. The test dataset can now be compared with training dataset in a very simple and effective, efficient way. The Proposed approach concentrates on using the Gaussian function based distance along with the k-Means instead of conventional distance function used by k-Means algorithm.

Table 1: Process system call matrix

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	Class
P_1	0	1	0	0	1	1	0	0	0	1	Normal
P_2	0	0	0	0	0	1	1	1	0	0	Normal
P_3	0	0	0	0	0	0	1	0	0	0	Normal
P_4	0	0	1	0	1	1	1	1	0	1	Normal
P_5	0	0	0	1	0	1	0	0	1	0	Abnormal
P_6	1	1	1	0	0	1	0	0	1	0	Abnormal
P_7	1	1	1	1	0	1	0	1	0	0	Abnormal
P_8	1	0	1	1	0	1	0	0	0	0	Abnormal
P_9	1	1	1	1	0	0	0	0	0	0	Abnormal

Table 2: Initial Clusters

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Cluster-1	0	1	0	0	1	1	0	0	0	1
Cluster-2	0	0	0	0	0	1	1	1	0	0

Table 3: Similarity of Process with Initial Clusters

	Cluster-1	Cluster-2	Class
P_1	1	X	1
P_2	X	1	2
P_3	0.5091	0.6727	2
P_4	0.7195	0.7545	2
P_5	0.6493	0.6073	1
P_6	0.6493	0.5792	1
P_7	0.6493	0.6318	1
P_8	0.5792	0.5909	2
P_9	0.5792	0.5091	1

Table 4: Clusters

	Processes
Cluster-1	1,5,6,7,9
Cluster-2	2,3,4,8

Table 5: Similarity of Process with Initial Clusters

	Cluster-1	Cluster-2	Decision
P_1	0.7079	0.6998	1
P_2	0.6871	0.8510	2
P_3	0.6871	0.8090	2
P_4	0.6236	0.7669	2
P_5	0.7828	0.7327	1
P_6	0.8419	0.7135	1
P_7	0.8145	0.6798	1
P_8	0.7987	0.7248	1
P_9	0.7987	0.6624	1

Table 6: Clusters: STAGE-2

	Processes
Cluster-1	1,5,6,7,8,9
Cluster-2	2,3,4

Table 7: Similarity of Process with Initial Clusters: STAGE-3

	Cluster-1	Cluster-2	DECISION
P_1	0.6080	0.6587	1
P_2	0.6858	0.8005	2
P_3	0.6858	0.7236	2
P_4	0.6272	0.6852	2
P_5	0.7788	0.6965	1
P_6	0.8321	0.6491	1
P_7	0.8128	0.6572	1
P_8	0.8320	0.6677	1
P_9	0.8128	0.6234	1

Table 8: Clusters: STAGE-3

	Processes
Cluster-1	1,5,6,7,8,9
Cluster-2	2,3,4

Table 9: Final Clusters Formed

	Processes
Cluster-1	1,5,6,7,8,9
Cluster-2	2,3,4

Table 10: NN, Cluster distances, Neighbor Distances w.r.t each process

	Similarity Value w.r.t, Clusters generated		Cluster Allotment	Nearest Neighbor	Similarity (Process, NN)
	Cluster-1	Cluster-2			
P_1	0.6880	0.6587	1	P_6	0.6491
P_2	0.6858	0.8005	2	P_4	0.7624
P_3	0.6858	0.7236	2	P_2	0.6832
P_4	0.6272	0.6852	2	P_2	0.7624
P_5	0.7788	0.6965	1	P_7	0.7195
P_6	0.8321	0.6491	1	P_7	0.859
P_7	0.8128	0.6572	1	P_6	0.859
P_8	0.8320	0.6677	1	P_9	0.8026
P_9	0.8128	0.6234	1	P_8	0.8026

Table 11: Nearest Neighbors for Processes in Cluster 1

	Similarity					
	P1	P5	P6	P7	P8	P9
P1	X	0.5906	0.6491	0.6174	0.5785	0.5769
P5	0.5906	X	0.6495	0.7195	0.7082	0.5923
P6	0.6491	0.6495	X	0.859	0.7568	0.7551
P7	0.6174	0.7195	0.859	X	0.7901	0.7886
P8	0.5785	0.7082	0.7568	0.7901	X	0.8026
P4	0.5769	0.5923	0.7551	0.7886	0.8026	X

Table 12: Nearest Neighbors for Processes in Cluster-2

	Similarity			Nearest Neighbor
	P2	P3	P4	
P2	X	0.6832	0.7624	P4
P3	0.6858	X	0.6040	P2
P4	0.7624	0.6040	X	P2

5 Case Study

The table.1 shows the process system call matrix and the corresponding class label for each process. The last column of table.1 corresponds to the class label. Here, we have two classes called normal and abnormal [Lin, 2015] [Liao, 2002].

So, we choose to cluster these records in to two clusters, cluster-1 and cluster-2. For this, we use k-Means clustering algorithm as we can specify the number

Table 13: Calculation of Total Similarity and Normalized Similarity

Process	Similarity Value w.r.t. Clusters generated		Clusters Generated	NN	Sim(NN)	Total Similarity Value	Norm TotalSim = Sim / 3
	Sim-C1	Sim-C2				Sim=Sim.C1+ Sim.C2+ NNSim	
P_1	0.6880	0.6587	1	P_6	0.6491	1.9958	0.665267
P_2	0.6858	0.8005	2	P_4	0.7624	2.2487	0.749567
P_3	0.6858	0.7236	2	P_2	0.6832	2.0926	0.697533
P_4	0.6272	0.6852	2	P_2	0.7624	2.0748	0.6916
P_5	0.7788	0.6965	1	P_7	0.7195	2.1948	0.7316
P_6	0.8321	0.6491	1	P_7	0.859	2.3402	0.780067
P_7	0.8128	0.6572	1	P_6	0.859	2.329	0.776333
P_8	0.8320	0.6677	1	P_9	0.8026	2.3023	0.767433
P_9	0.8128	0.6234	1	P_8	0.8026	2.2388	0.746267

Table 14: Processes after Dimensionality Reduction using proposed measure

Total Process Set	Similarity Value	Distance
P_1	0.665267	0.334733
P_2	0.749567	0.250433
P_3	0.697533	0.302467
P_4	0.6916	0.3084
P_5	0.7316	0.2684
P_6	0.780067	0.219933
P_7	0.776333	0.223667
P_8	0.767433	0.232567
P_9	0.746267	0.253733

Table 15: New test process with NN, Similarity and Distance Values

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Nearest NN	Sim	Dist
P_{test}	0	1	0	0	1	1	0	0	0	1	Process-1	1.0	0
P_{new}	1	1	1	1	0	1	0	1	1	0	Process-7	1.0	0

Table 16: Classifying New Test Process for Intrusion

	Nearest NN	Sim Dist
P_{test}	Process-1	Normal
P_{new}	Process-7	Abnormal

of clusters required by specifying the value of k=2.

But, the difference lies in the distance measure used. Here, we use the Gaussian based similarity measure for clustering using k-Means as against to traditional distance measures used such as Euclidean, Cosine, City block.

At the end of the clustering process, we have two clusters as shown in Table.2 We perform 3 iterations using k-Means by recording the clusters at each iteration. We terminate the process of clustering, when the clusters formed for two successive stages remain same. This process is shown for each iteration using the Table. 4, through Table.9.

6 Conclusion

In this work, the main contribution is in defining the similarity measure which has finite lower and upper bounds. The measure designed is Gaussian function based distance measure. The k-Means algorithm is chosen for clustering using the proposed distance measure to cluster both the training and testing data samples. The training and test datasets are transformed to single dimensional feature with the use of k-Means and the proposed distance measure. The significance of the proposed distance measure is that, it considers the distribution of the system calls behaviour over the entire training samples. This makes the computation accurate, even in binary form. The similarity value lies between 0 and 1.

References

[Abadeh, 2011] Mohammad Saniee Abadeh, Hamid Mohamadi, JafarHabibi: Design and analysis of genetic fuzzy systems for intrusion detection in computer networks, Elsevier Journal of Expert Systems with Applications 38, (2011) Pages 7067-7075

[Aljawarneh, 2016] Shadi A. Aljawarneh, Raja A. Moftah, Abdelsalam M. Maatuk, Investigations of automatic methods for detecting the polymorphic worms signatures, Future Generation Computer Systems, Volume 60, July 2016, Pages 67-77, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2016.01.020.

[Aljawarneh, 2010] Shadi Aljawarneh, Faisal Alkhateeb, and Eslam Al Maghayreh. 2010. A semantic data validation service for web applications. J. Theor. Appl. Electron. Commer. Res. 5, 1 (April 2010), 39-55.

[Barbar, 2001] Daniel Barbar, Julia Couto, Sushil Jajodia and Ningning Wu.: ADAM: a testbed for exploring the use of data mining in intrusion detection. SIGMOD Rec. 30, 4 (December 2001), 15-24.

[Govindarajan, 2011] M. Govindarajan, RM. Chandrasekaran: Intrusion detection using neural based hybrid classication methods, Elsevier, Computer Networks 55 (2011) 16621671.

[Hsu, 2011] Hsu, W. F. (2011). A server side solution to prevent information leakage by cross site scripting attack.

[Hubballi, 2014] Neminath Hubballi and Vinoth Suryanarayanan. 2014. Review: False alarm minimization techniques in signature-based intrusion detection systems: A survey. Comput. Commun. 49 (August 2014), 1-17. DOI=http://dx.doi.org/10.1016/j.comcom.2014.04.012

[Hyun, 2003] Sang Hyun Oh and Won Suk Lee: An anomaly intrusion detection method by clustering normal user behaviour, Computers & Security, Volume 22, Issue 7, October 2003, Pages 596612

[Joachims, 1999] Thorsten Joachims: Making large-scale support vector machine learning practical. In Advances in kernel methods, Bernhard Schlkopf, Christopher J. C. Burges, and Alexander J. Smola (Eds.). MIT Press, Cambridge, MA, USA 169-184.

[Krsul, 1998] Ivan Victor Krsul, a thesis report on Software vulnerability analysis, Purdue University.

[Kumar1, 2015] Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narasimha: Intrusion Detection Using Text Processing Techniques: A Recent Survey. In Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). ACM, New York, NY, USA, Article 55, 6 pages. DOI=http://dx.doi.org/10.1145/2832987.2833067.

[Kumar2, 2015] Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narasimha. : An approach for Intrusion Detection using Text Mining Techniques. In Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). ACM, New York, NY, USA, Article 63, 6 pages. DOI=http://dx.doi.org/10.1145/2832987.

[Kumar3, 2015] Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narasimha. 2015. An improved k-Means Clustering algorithm for Intrusion Detection using Gaussian function. In Proceedings of the The International Conference on Engineering & MIS 2015 (ICEMIS '15). ACM, New York, NY, USA, , Article 69 , 7 pages. DOI=http://dx.doi.org/10.1145/2832987.2833082

[Kumar4, 2016] Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narasimha. : Intrusion Detection A Text Mining Based Approach. Special issue on Computing Applications and Data Mining International Journal of Computer Science and Information Security (IJCSIS), Vol. 14 S1, February 2016 (pp. 76-88)

[Kumar5, 2016] Gunupudi Rajesh Kumar, Mangathayaru Nimmala, G. Narasimha. : A Novel Similarity Measure for Intrusion Detection using Gaussian Function. Technical Journal of the Faculty of Engineering, Vol.39 No.2, (pp. 173-183) 2016

[Lee, 1998] Wenke Lee and Salvatore J. Stolfo : Data mining approaches for intrusion detection. In Proceedings of the 7th conference on USENIX Security Symposium -Volume 7 (SSYM98), Vol. 7. USENIX Association, Berkeley, CA, USA, 6-6.

[Liao, 2002] Yihua Liao, V. Rao Vemuri: Using Text Categorization Techniques for Intrusion Detection, Proceedings of the 11th USENIX Security Symposium, Pages 51-59 USENIX Association Berkeley, CA, USA 2002

[Lin, 2015] Wei-Chao Lin, Shih-Wen Ke, Chih-Fong Tsai: CANN: An intrusion detection system based on combining cluster enters and nearest neighbours, Knowledge-Based Systems 78 (2015) 13-21

[Manganaris, 1999] Stefanos Manganaris, Marvin Christensen, Dan Zerkle, and Keith Hermiz. 2000. A data mining analysis of RTID alarms. Comput. Netw. 34, 4 (October 2000), 571-577. DOI=http://dx.doi.org/10.1016/S1389-1286 (00)00138-9

[Meng, 2013] Yuxin Meng, Wenjuan Li, and Lam-For Kwok. Towards adaptive character frequency-based exclusive signature matching scheme and its applications in distributed intrusion detection. Comput. Netw. 57, 17 (December 2013), 3630-3640.

[Modi, 2013] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttukrishnan Rajarajan : Journal of Network and Computer Applications, A survey of intrusion detection techniques in Cloud, Volume 36, Issue 1, Pages 4257, 2013

[Nguyen, 2010] Viet Hung Nguyen and Le Minh Sang Tran. 2010. Predicting vulnerable software components with dependency graphs. In Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec '10). ACM, New York, NY, USA, , Article 3 , 8 pages. DOI=http://dx.doi.org/10.1145/1853919.1853923

[Pistoia, 2015] Pistoia, M., Segal, O., & Tripp, O. (2015). U.S. Patent No. 8984642. Washington, DC: U.S. Patent and Trademark Office.

https://www.google.ch/patents/US8984642

[Sharma, 2007] Alok Sharma, Arun K Pujari, Kuldip K Paliwal: Intrusion Detection using text processing techniques with a kernel based similarity measure, Elsevier Journal of Computers and Security, Pages. 488-495, 2007

[Stolfo, 1999] Lee, W. Stolfo, S. Kui, M.: A Data Mining Framework for Building Intrusion Detection Models. IEEE Symposium on Security and Privacy (1999) 120-132

[Treinen, 2006] James J. Treinen and Ramakrishna Thurimella: A framework for the application of association rule mining in large intrusion detection infrastructures. In Proceedings of the 9th international conference on Recent Advances in Intrusion Detection (RAID'06), Diego Zamboni and Christopher Kruegel (Eds.). Springer-Verlag, Berlin, Heidelberg, 1-18.

[Vapnik, 1995] Vapnik, Vladimir: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA, Volume 6592 of the series Lecture Notes in Computer Science pp 353-362.