Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 125 (2018) 709–716

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India

Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning

Abhishek Verma[a,*], Virender Ranga[a]

[a]*Department of Computer Engineering, NIT Kurukshetra, India*

Abstract

A lot of research is being done on the development of effective Network Intrusion Detection Systems. Anomaly based Network Intrusion Detection Systems are preferred over Signature based Network Intrusion Detection Systems because of their better significance in detecting novel attacks. The research on the datasets being used for training and testing purpose in the detection model is equally concerned as better dataset quality can advance offline Intrusion Detection. Benchmark datasets like KDD99 and NSL-KDD cup 99 are outdated and face some major issues, which make them unsuitable for evaluating Anomaly based Network Intrusion Detection Systems. This paper presents the statistical analysis of labelled flow based CIDDS-001 dataset using *k*-nearest neighbour classification and *k*-means clustering algorithms. The analysis is done with respect to some prominent evaluation metrics used for evaluating Network Intrusion Detection Systems including Detection Rate,Accuracy and False Positive Rate.

Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications

Keywords: Anomaly, Signature, Datasets, Labelled flow, *k*-nearest neighbour classification, *k*-means clustering, Analysis, Metrics

1. Introduction

Network security has become one of the most concerning problems for internet users and service providers with drastic increase in the internet usage [1]. A secure network can be defined in terms of its hardware and software protection against various intrusions. A network can be secured by implementing a resilient monitoring, analysis and defence mechanisms. Network Intrusion detection systems (NIDS) [2] forms a class of systems which implement these mechanisms in order to defend a network from insider and outsider intrusions. These systems monitor the incoming and outgoing traffic in a network, perform time to time analysis and report when some intrusion is detected. NIDS can be broadly categorised into Misuse detection (MD) [3], Anomaly Detection (AD) [4]. MD based NIDS use signatures or patterns of already existing attacks to detect intrusions. While AD based NIDS check for strict deviations from the normal profiles of the network traffic and report it as attack. MD based NIDS have a fast Detection Rate (DR) with less False Positive Rate (FPR) as compared to AD. However AD based NIDS are able to detect novel attacks over networks and this property outperforms them over MD based NIDS. MD works on the offline data whereas AD works better on the online data. Machine Learning (ML) [5] is playing a major role in the development of better NIDS.

* Corresponding author
E-mail addresses: abhishek_6170034@nitkkr.ac.in (Abhishek Verma)., virender.ranga@nitkkr.ac.in (Virender Ranga).

1877-0509
Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications
10.1016/j.procs.2017.12.091

It involves a system to learn from the recorded traffic patterns or signatures and then act accordingly for upcoming traffic patterns. Training and testing are the two major task involved in the ML. ML requires a large and complex datasets comprising of different types of normal and abnormal traffic patterns. There is also a need of applying ML algorithms to NIDS which have a less computational time and space complexity for a better learning. In this work we have analysed CIDDS-001 dataset using some prominent NIDS evaluation metrics like DR, FPR, Accuracy, Precision and F-measure [6]. We have used distance-based ML models [7] like *k*NN classification algorithm [8] due to its better DR and *k*-means clustering [9] due to its fast execution time.

1.1. CIDDS-001 Dataset

CIDDS-001 (Coburg Network Intrusion Detection Dataset) [10] is a labelled flow [11] based dataset developed for the evaluation of Anomaly based NIDS. This dataset contains unidirectional NetFlow data. It consists of traffic data from two server's i.e. OpenStack and External server. The dataset is generated by emulating small business environment which consist of OpenStack environment having internal servers (web, file, backup and mail) and an External Server (file synchronization and web server) which is deployed on the internet to capture real and up-to-date traffic from the internet. The dataset consists of three logs files (attack logs, client configurations and client logs) and traffic data from two servers where each server traffic comprises of 4 four week captured traffic data. The CIDDS-001 has 14 attributes out of which 12 have been used in this empirical study. This dataset consists large number of traffic instances out of which 153026 instances from External Server and 172839 instances from OpenStack Server been used for the analysis. Table 1 provides the description of CIDDS-001 dataset attributes .

Table 1: Classwise detail of CIDDS-001 dataset attributes

Sl. no.	Attribute Name	Attribute Description
1	Src IP	Source IP Address
2	Src Port	Source Port
3	Dest IP	Destination IP Address
4	Dest Port	Destination Port
5	Proto	Transport Protocol (e.g. ICMP, TCP, or UDP)
6	Date first seen	Start time flow first seen
7	Duration	Duration of the flow
8	Bytes	Number of transmitted bytes
9	Packets	Number of transmitted packets
10	Flags	OR concatenation of all TCP Flags
11	Class	Class label (Normal, Attacker, Victim, Suspicious and Unknown)
12	AttackType	Type of Attack (PortScan, DoS, Bruteforce, PingScan)
13	AttackID	Unique Attack id. Allows attacks which belong to the same class carry the same attack id
14	AttackDescription	Provides additional information about the set attack parameters (e.g. the number of attempted password guesses for SSH-Brute-Force attacks)

1.2. Objective

The objective of this research work is to perform analysis of the CIDDS-001 dataset from the ML point of view. In this work we study how the classification and clustering algorithms perform on the dataset considering 12 important attributes. Our objectives include classifying and clustering Network traffic of OpenStack and External Servers into $Normal, Attacker, Victim, Suspicious$ and $Unknown$ classes. We have used some prominent metrics for evaluating *k*NN classifier and shown Confusion Matrix generated from *k*-means clustering.

2. Related Work

In [12] the class wise analysis of NSL-KDD [13] dataset is performed. The attributes of the dataset are categorized into four classes i.e. $Basic, Content, Traffic$ and $Host$. Contribution of every class is evaluated in terms of DR and FAR. Siddiqui et al. [14] performed the analysis of NSL-KDD dataset for Intrusion Detection (ID) using clustering algorithm based Data Mining techniques. They used k-means clustering to build 1000 clusters over 494020 records and focused on building relationship between attack types and protocols used to make intrusion. Artificial neutral network (ANN) is used in [15] for the analysis of NSL-KDD dataset. DR for intrusion detection and attack type classification was found to be 81.2% and 72.9% respectively. In [16] the study of irrelevant features of KDD99 [17]and UNSW-NB15 [18] which leads to efficiency reduction of NIDS is done. An Association Rule Mining algorithm is used for strongest feature selection from the two datasets and then classifiers are used for evaluation in terms of accuracy and False Alarm Rate (FAR). In results it is shown that features of UNSW-NB15 are much efficient than KDD99 dataset. Kayacik et al.[19] studied three Intrusion Detection System (IDS) benchmark datasets using ML algorithms. Clustering and Neural Network algorithms are used to analyse the IDS datasets and find the differences between synthetic and real world traffic.

3. Experimental Setup

3.1. Research Methodology

Following steps are followed as part of the research methodology:

- CIDDS-001 dataset is selected due to the necessity of Flow-based benchmark data sets for Anomaly based NIDS.
- Simulation is performed on popular Data Mining tool Weka.
- kNN classifier is used for multi-class classification on Weka. It classifies instances as normal, attacker, victim, suspicious and unknown.
- k-means clustering is used for clustering the instances in above mentioned classes.
- Pre-processing of training and testing data files with 12 attributes is performed.
- Classification and clustering algorithms are simulated and results are tabulated.

3.2. Weka

Weka (Waikato Environment for Knowledge Analysis) [20] is open source data mining tool available under GNU General Public License. Weka is collection of ML algorithms used for performing data mining tasks. It consists various tools for dataset pre-processing, classification, regression, clustering, association rules and visualization. Weka version 3.9 is used in this work. Commonly uses data file formats like comma separated (csv) and attribute file format ($arff$) are supported by Weka. For the simulation, already available csv file with 14 attributes is selected and csv file with 12 attributes is created through pre-processing tool of Weka.

3.3. k-nearest neighbour classifier

kNN classifier is instance based learning and classification algorithm which is also known as lazy classifier. It is most commonly used distance-based classifier and uses each training instance as a prototypical instance. This classifier is built on distance function that basically measures the similarity or difference between two instances. The standard distance measure i.e. Euclidean distance $D(p, q)$ between two instances p and q is defined as Equation 1 [21].

$$D(p,q) = \sqrt{\sum_{i=1}^{r}(p_i - q_i)^2} \tag{1}$$

Where p_i is the i^{th} featured element if the instance p, q_i is the i^{th} featured element of the instance q and r is the total number if the features in the dataset. Consider that the design set for kNN classifier is Z. Let M is total number of samples in Z. Let $N = \{N_1, N_2, N_3, \ldots, N_O\}$ are the O distinct class labels that are available in M. Consider p be an input vector for which the class label need to be predicted. Let q_i denotes the i^{th} vector in the design set M. kNN classification algorithm finds k closest vectors in the design set M to the input vector p. Then the input vector p is classified to class N_j if the majority of the k closest vectors have their class as N_j.

3.4. k-means clustering

In a distance-based perspective, unsupervised learning is generally referred to clustering and k-means is known to be one of the simplest among those algorithms. k-means clustering partitions n instances into k clusters in which each instance is associated to the cluster with nearest mean. It has a very loose relationship with kNN classifier. Given a set of instances $(p_1, p_2, \ldots, p_n)$, where each instance is a d-dimensional real vector. k-means clustering aims to partition p instances into $k(\leq p)$ sets $Z = \{Z_1, Z_2, \ldots, Z_k\}$ in order to minimize the within cluster sum of squares (i.e. variance). Mathematically it can be illustrated as Equation 2 [22].

$$arg_z min \sum_{i=1}^{k} \sum_{p \epsilon Z_i}^{max} \|p - \mu_i\|^2 = arg_z min \sum_{i=1}^{k} |Z_i| Var\, Z_i \tag{2}$$

Where μ_i is the mean of points in Z_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster subsequently because the total variance is constant, it is similar to maximizing the squared deviations between points in different clusters. k-Means Clustering is fast, robust and easy to implement.

3.5. Evaluation Metrics

Performance of Network Intrusion Detection Systems (NIDS) is evaluated using some prominent metrics [6]. These metrics include Detection Rate (DR), False Positive rate (FPR), F-measure, Accuracy and Precision. Complexity of any dataset is measured on the basis of FPR and Accuracy. All these metrics are evaluated from the elements of the classification metrics. Let the elements of classification metrics are $EC_{KNN,KMC} = \{FP, FN, TP, TN\}$ where FP (false positive) is the number of instances misclassified as attacks, FN (false negative) is the number of instances misclassified as non-attacks, TP (true positive) is the number of instances correctly classified as attacks and TN (true negative) is the number of instances correctly classified as non-attacks. The accuracy is the ratio of correctly classified instances to all the instances, whether correctly or incorrectly classified and denoted by Equation 3.

The Detection Rate (true positive rate) is the ratio of all the correctly classified instances as attacks to all the correctly classified attacks and misclassified non-attacks. DR is denoted by Equation 4. Precision (positive predictive value) is the ratio of correctly classified instances as attacks to total of correctly classified instances as attacks and misclassified instances as attacks. Equation 5 represents Precision in terms of classification metrics. F-measure is a harmonic mean of precision and detection rate or true positive rate and denoted by Equation 6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Detection\ Rate(DR) = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F - measure = \frac{2TP}{2TP + FP + FN} \tag{6}$$

A good IDS should have high accuracy and DR but FPR should be low. FAR(false alarm rate) is directly proportional to the misclassification rate.

4. Experimental Results and Discussion

The examination has been done using distance-based ML models to statistically analyse the complexity of CIDDS-001 data set. This study uses two ML techniques for analysis i.e. *k*NN classification and *k*-means clustering. The simulation is done on Weka (version 3.9) using Intel(R) 7700 having clock speed of 3.60 GHz processor with 8 GB memory. For each pre-processed dataset file 66% of pre-processed data is used for creating model or training and rest 34% for testing.

4.1. Analysis using k-nearest neighbour classifier

First *k*NN classification is used for analysis of External Server traffic data. Week 3 External Server traffic data file consisting of total 153026 instances is pre-processed and converted to dataset having 12 features. We have neglected AttackID and AttackDescription features because they just give more information about executed attacks .Hence these attributes do not play any important role in classification. Performance of *k*NN classifier is shown with 1-Neighbour, 2-Neighbours, 3-Neighbours, 4-Neighbours and 5-Neighbours in Table 2. For every execution on External traffic data, *k*NN classifier performance is evaluated on the basis of correctly classified traffic instance into *suspicious*, *unknown*, *normal*, *attacker* and *victim* class. Secondly, *k*NN classifier is analysed over OpenStack Server

Table 2: Performance of *k*NN classifier on External Server traffic data

Neighbours	Evaluation Metrics					Class	Accuracy
	TP Rate	FP Rate	Precision	Detection Rate	F-measure		
1NN	0.995	0.004	0.998	0.995	0.996	suspicious	0.995
	0.993	0.004	0.986	0.993	0.990	unknown	
	1.000	0.000	0.999	1.000	0.999	normal	
	1.000	0.000	1.000	1.000	1.000	attacker	
	1.000	0.000	1.000	1.000	1.000	victim	
2NN	0.997	0.006	0.997	0.997	0.997	suspicious	0.996
	0.990	0.003	0.991	0.990	0.990	unknown	
	1.000	0.000	0.999	1.000	1.000	normal	
	1.000	0.000	1.000	1.000	1.000	attacker	
	1.000	0.000	1.000	1.000	1.000	victim	
3NN	0.994	0.006	0.997	0.994	0.995	suspicious	0.994
	0.991	0.005	0.983	0.991	0.987	unknown	
	1.000	0.000	0.996	1.000	0.998	normal	
	1.000	0.000	1.000	1.000	1.000	attacker	
	1.000	0.000	0.996	1.000	0.998	victim	
4NN	0.996	0.007	0.996	0.994	0.996	suspicious	0.995
	0.989	0.003	0.988	0.989	0.988	unknown	
	1.000	0.000	0.996	1.000	0.998	normal	
	1.000	0.000	1.000	1.000	1.000	attacker	
	1.000	0.000	1.000	1.000	1.000	victim	
5NN	0.993	0.006	0.996	0.993	0.995	suspicious	0.993
	0.989	0.005	0.982	0.989	0.986	unknown	
	1.000	0.000	0.996	1.000	0.998	normal	
	1.000	0.000	1.000	1.000	1.000	attacker	
	1.000	0.000	1.000	1.000	1.000	victim	

traffic data. We have selected random 172839 instances from week 1 traffic data using Reservoir Sampling [23]. Performance of *k*NN classifier is shown in Table 3. For every execution on External traffic data, kNN classifier performance is evaluated on the basis of correctly classified traffic instance into *victim*, *normal*, and *attacker*. Approximately for every execution of *k*NN classifier on the External Server traffic data, models average accuracy is 99%. Maximum accuracy of 99.6% is achieved with 2NN and minimum 99.3% with 5NN. Similarly for *k*NN classifier execution on OpenStack traffic data models average accuracy is 100% in each case, this may be due to random sampling of instances from the dataset file which can lead to some biased instance selections. Number of Neighbours can be increased for much better dataset analysis. Feature selection algorithms can be used to select most strong features in the dataset in order to further enhance the classification accuracy. Dataset can be analysed on other evaluation metrics like ROC curve [24] and FAR also. Statistical analysis of CIDDS-001 can be done using other ML algorithms in order to analyse their performance on the dataset.

Table 3: Performance of *k*NN classifier on OpenStack Server traffic data

Neighbours	Evaluation Metrics					Class	Accuracy
	TP Rate	FP Rate	Precision	Detection Rate	F-measure		
1NN	1.000	0.000	1.000	1.000	1.000	victim	1.000
	1.000	0.001	1.000	1.000	1.000	normal	
	0.999	0.000	1.000	0.999	1.000	attacker	
2NN	1.000	0.000	1.000	1.000	1.000	victim	1.000
	1.000	0.001	1.000	1.000	1.000	normal	
	0.999	0.000	1.000	0.999	0.999	attacker	
3NN	1.000	0.000	1.000	1.000	1.000	victim	1.000
	1.000	0.001	1.000	1.000	1.000	normal	
	0.999	0.000	1.000	0.999	0.999	attacker	
4NN	1.000	0.000	1.000	1.000	1.000	victim	1.000
	1.000	0.001	1.000	1.000	1.000	normal	
	0.998	0.000	1.000	0.998	0.999	attacker	
5NN	1.000	0.000	1.000	1.000	1.000	victim	1.000
	1.000	0.001	1.000	1.000	1.000	normal	
	0.999	0.000	1.000	0.999	1.000	attacker	

4.2. Analysis using k-means clustering

Firstly, *k*-means clustering is used for analysis of External Server traffic data. Week 1 External Server traffic data file consisting of total 153026 instances is selected for pre-processing and converted to dataset having 12 features. Feature "class" (*suspicious*, *unknown*, *normal*, *attacker* and *victim*) of the dataset is set as the class of Clusters. Table 4 shows the Multi-class Confusion matrix for this experiment. In this experiment 94710 (61.8914%) instances are incorrectly clustered. Within cluster sum of squared errors (CSS) calculated is 449172.438. Secondly, *k*-means clustering is used over OpenStack Server traffic data. 150000 instances from week 1 traffic data are selected using Reservoir Sampling

Table 4: Confusion Matrix for *k*-means clustering on External Server traffic data

k-Means External Server		Predicted Class				
		suspicious	unknown	normal	attacker	victim
Actual Class	suspicious	28952	3788	28061	17218	19833
	unknown	1977	14045	330	2545	14940
	normal	32	20	3038	32	3058
	attacker	0	4	719	8532	0
	victim	2153	0	0	0	3749

then pre-processed and converted to a dataset having 12 features. Feature "class" (*victim*, *attacker* and *normal*) of the dataset is set as the class of clusters. Table 5 shows Multi-class Confusion matrix for second experiment. In this experiment 506 (0.3373%) instances are incorrectly clustered and calculated within CSS is 313480.297. Different clustering techniques (Hierarchical, Fuzzy-Means, *k*-medoids clustering etc.) of ML can be employed with different distance measures (Manhattan, Jaccard and Cosine etc.) for much better analysis of the dataset.

5. Conclusion and Future Work

We have discussed the statistical analysis and evaluation of labelled flow based CIDDS-001 dataset used for evaluating Anomaly based Network Intrusion Detection Systems in this paper. Two techniques, *k*-nearest neighbour classification and *k*-means clustering are used to measure the complexity in terms of prominent metrics. On the basis of evaluation results it can be concluded that both *k*-nearest neighbour classification and *k*-means clustering perform well over CIDDS-001 datset in terms of used prominent metrics. Hence the dataset can be used for the evaluation of Anomaly based Network Intrusion Detection Systems. In future, we have planned to do a comparative study of CIDDS-001 dataset with existing Network Intrusion Detection Systems benchmarking datasets in order to study the complexity of this dataset over other datasets.

References

[1] Medaglia, C. M., & Serbanati, A. (2010). An overview of privacy and security issues in the internet of things. In The Internet of Things (pp. 389-395). Springer, New York, NY.

[2] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. Computer Networks, 31(8), 805-822.

[3] Zhengbing, H., Zhitang, L., & Junqi, W. (2008, January). A novel Network Intrusion Detection System (NIDS) based on signatures search of data mining. In Proceedings of the 1st international Conference on Forensic Applications and Techniques in Telecommunications, information, and Multimedia and Workshop (p. 45). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[4] Garcia-Teodoro, P., Diaz-Verdejo, J., MaciÃą-FernÃąndez, G., & VÃązquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1), 18-28.

[5] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 305-316). IEEE.

[6] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D.,& Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings (Vol. 2, pp. 12-26). IEEE.

[7] Flach, P. (2012). Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press.

[8] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[9] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

[10] CIDDS-001 dataset. (2017, Aug.) [Online] Available: https://www.hs-coburg.de/forschung-kooperation/forschungsprojekte-oeffentlich/ingenieurwissenschaften/cidds-coburg-intrusion-detection-data-sets.html

[11] Ring, M., Wunderlich, S., Grudl, D., Landes, D., Hotho, A.(2017). Flow-based benchmark data sets for intrusion detection. In Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), in press. ACPI

[12] Aggarwal, P., & Sharma, S. K. (2015). Analysis of KDD dataset attributes-class wise for intrusion detection. Procedia Computer Science, 57, 842-851.

[13] NSL-KDD dataset. (2017, Aug.) [Online] Available: http://iscx.info/NSL-KDD/27. http://www.cs.waikato.ac.nz/âĹijml/weka/.

[14] Siddiqui, M. K., & Naahid, S. (2013). Analysis of KDD CUP 99 dataset using clustering based data mining. International Journal of Database Theory and Application, 6(5), 23-34.

[15] Ingre, B., & Yadav, A. (2015, January). Performance analysis of NSL-KDD dataset using ANN. In Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on (pp. 92-96). IEEE.

Table 5: Confusion Matrix for *k*-means clustering on OpenStack Server traffic data

k-Means OpenStack		Predicted Class		
		victim	attacker	normal
Actual Class	victim	57955	0	0
	attacker	0	57963	0
	normal	90	416	33576

[16] Moustafa, N., & Slay, J. (2015, November). The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems. In Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on (pp. 25-31). IEEE.
[17] KDD Cup 1999. (2014, Nov.) [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/.
[18] UNSW-NB15 dataset. (2017, Aug.). [Online]. Available: https: //www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/
[19] KayacÄśk, H. G., & Zincir-Heywood, N. (2005, May). Analysis of three intrusion detection system benchmark datasets using machine learning algorithms. In International Conference on Intelligence and Security Informatics (pp. 362-367). Springer, Berlin, Heidelberg.
[20] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.
[21] Kaur, D. (2014). A Comparative Study of various Distance Measures for Software fault prediction. arXiv preprint arXiv:1411.7474.
[22] Kriegel, H. P., Schubert, E., & Zimek, A. (2016). The (black) art of runtime evaluation: Are we comparing algorithms or implementations?. Knowledge and Information Systems, 1-38.
[23] Vitter, J. S. (1985). Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS), 11(1), 37-57.
[24] Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. Machine learning, 77(1), 103-123.