Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 928–936

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Intrusion Detection in Computer Networks using Lazy Learning Algorithm

Aditya Chellam[a], Ramanathan L[a], Ramani S[a]

[a]*School of Computer Science and Engineering, VIT, Vellore 632014, India*

Abstract

Intrusion Detection Systems (IDS) are used in computer networks to safeguard the integrity and confidentiality of sensitive data. In recent years, network traffic has become sizeable enough to be considered under the big data domain. Current machine learning based techniques used in IDS are largely defined on eager learning paradigms which lose performance efficiency by trying to generalize training data before receiving queries thereby incurring overheads for trivial computations. This paper, proposes the use of lazy learning methodologies to improve overall performance of IDS. A novel heuristic weight based indexing technique has been used to overcome the drawback of high search complexity inherent in lazy learning. IBk and LWL, two popular lazy learning algorithms have been compared and applied on the NSL-KDD dataset for simulating a real-world like scenario and comparing their relative performances with hw-IBk. The results of this paper clearly indicate lazy algorithms as a viable solution for real-world network intrusion detection.

Keywords: Lazy Learning; Intrusion Detection System; Machine Learning; IBk; kNN

*Corresponding Author: aditya.chellam2015@vit.ac.in

10.1016/j.procs.2018.05.108

1. Introduction

The predominant strategy for observing systems for vindictive movement or information infringement is the utilization of Intrusion Detection System (IDS). Any identified approach of infringement is ordinarily revealed either to an overseer or accumulated midway utilizing a Security Information and Event Management (SIEM) framework. A SIEM framework system-cluster comes about because of numerous sources and makes utilization of preventive sifting procedures to decide the validity of identified assault.Network Intrusion Detection Systems (NIDS) are strategically positioned and demonstrate the framework screen motion between all nodes on the framework. It supervises the actions on the entire network and unusual subnet activities are corresponded to a library of assaults that are already known.Once an assault is recognized, or irregular conduct is detected, the caution can be sent to the administrator. A case of an NIDS would introduce it on the subnet where firewalls are situated, so as to check whether somebody is attempting to break into the firewall. In a perfect world, one would check all inbound and outbound activity; however, doing as such, may make a bottleneck that would weaken the general speed of the system. OPNET and NetSim are regularly utilized instruments for reproducingnetwork intrusion discovery frameworks. NID Systems are additionally equipped for contrasting marks for comparative bundles with connection and drop unsafe distinguished parcels which have a mark coordinating the records in the NIDS.NIDS can be characterizedinto two subgroups based on the intuitiveness of the framework, namely, disconnected and online NIDS. Disconnected NIDS detect assaults by passing the information through a set ofprocedures[6]. In the case of Online NIDS, Ethernet bundles are scrutinized and tenets are applied to detect assaults.

2. Data Mining in Computer Networks

Data mining techniques for intrusion detection are chiefly based on follows –

- Frequent pattern mining
- Classification
- Clustering
- Mining data streams

Data mining in the network security context is defined as the non-trivial process of identifying verified and important data by characterizing the underlying patterns in the networks.Machine Learning based data mining techniques have tremendous applications in detecting underlying patterns in network traffic data. Supervised learning is performed to learn accurate and exact models from previous intrusion logs.Alternatively, in unsupervised learning, suspicious activities are detected and subsequently identified.

2.1. Lazy Learning Algorithms

IBk Classifier – In the K-nearest neighbour's classifier, predictions are made based on the relative node distances of instances from each class. There is no fixed value of K suitable for all domains, and the algorithm uses cross validation of K in order to pick an appropriate value.

LWL Classifier – Locally Weighted Learning (otherwise called memory-based learning, case-based learning, lazy-learning, and firmly identified with kernel density estimation, similitude seeking and case-based thinking). Locally Weighted Learning is basic, yet, engaging, both naturally and measurably. When you need to foresee, what will occur later on, you basically venture into a database of all your past encounters, get some comparative encounters, join them (maybe by a weighted normal that weights more comparative encounters all the more unequivocally) and utilize the blend to make an expectation, do a relapse, or numerous other more complex operations. The algorithm is extremely adaptable and provides a precise model in the long run.

2.2. Advantage of using lazy learning

The fundamental preferred standpoint picked up in utilizing a lazy learning strategy, for example, case-based thinking, is that the objective capacity will be locally approximated. Since the objective capacity is approximated locally for each question to the framework, lethargic learning frameworks can at the same time take care of numerous issues and arrangement effectively with changes in the issue area.

2.3. Simulation of real world network

The NSL-KDD dataset is recommended for this study as it takes care of a portion of the characteristic issues of the KDD'99 informational index as mentioned in. In spite of the way that, this new type of the KDD dataset still encounters a bit of the issues discussed by McHugh and may not be a faultless illustrative of existing veritable frameworks, in perspective of the lack of open source data indexes for framework based IDSs, it can in any case be reliably associated as an effective benchmark instructive record to enable investigators to analyse changed interference acknowledgment procedures[2].

Also, the NSL-KDD contains a sizeable number of records[2]. This favoured angle influences it to run the examinations on the aggregate set without the need to discretionarily pick a tiny bit. In this manner, appraisal eventual outcomes of different research work are expected to be essentially indistinguishable. The salient features of the NSL-LDD that make it more desirable than its predecessors are as follows, The quantity of picked records from each issue level is comparable to the rate of records in the primary KDD dataset. Thus, the portrayal rates of unmistakable machine learning methodologies vary in a broader region, which makes it more capable to have a correct evaluation of different learning techniques. Both test and prepared set contain appropriate number of instances, thus investigations can be run on the entire set seamlessly. Therefore, assessment aftereffects of different research works will be consistent and nearly alike.

3. Literature Survey

Various Machine Learning (ML) algorithms were surveyed for determining the optimum data mining solution to detect intrusions in computer networks. The various surveyed work has been enlisted in tabular form below.

Table 1. Literature Survey Table

Sr. No.	Author Name	Domain Addressed	Description	Algorithm Used	Advantage
1	David A Cieslak et al.	Imbalance in Network Intrusion Datasets	Actual Notre Dame traffic analysed to detect imbalance in real time network intrusions. Using ROC analysis, it is shown that over-sampling by artificial generation of minority (intrusion) class outdo oversampling by imitation and RIPPER's loss ratio method [6].	RIPPER rule learning, ROC used for analysis	Clustering based approach more suitable for intrusion detection and can deliver added enhancement over just artificial generation of occurrences.
2	Wei Wang and Roberto Battiti	Network Intrusion Detection	Normal intrusion behaviour profiled founded on regular data for irregularity detection and models of each type of attack built based on attack data for intrusion recognition[7].	Principal Component Analysis; proposed profiling algorithm	Accurate identification and computationally efficiency model for real-time intrusion identification.

3	Jiong Zhang et al.	Network Intrusion Detection	KDD'99 experiment to detect network intrusions using Random Forest Algorithm. Proposed model improves detection performance of current Network Intrusion Detection Systems (NIDS)[4].	Random Forest Algorithm	Can detect unknown intrusions and low, false positive rate; overcomes shortcomings of anomaly and misuse detection.
4	Steven Noel and SushilJajodia	Complex Network Attacks	Graph based technique elucidates multiple-step attacks by matching rows and columns of the clustered adjacency matrix permitting attack influence/responses to be identified and prioritized based on the number of attack steps to victim machines, and allows attack origins to be determined[3].	Adjacency Matrix Clustering	places intrusion checkpoints in context of susceptibility based attack graphs, making false alarms ostensible thus making inference of missed detection possible
5	Corvera S. et al	Anomaly Detection	Data mining technique used to cluster networks to detect anomalies using kNN based learning.	k-NN Algorithm for anomaly detection	Efficient and effective anomaly detection in networks
6	Wenke Lee et al.	Building IDS models	Reviewing programs used to excerpt Extensive set of features describing each node in system. Data mining programs are used to accurately learn rules capturing the behaviour of interruptions and normal events[1].	Meta Classification RIPPER used for anomaly detection. Bro Engine used for packet filtering and reassembling.	Proposed model shows best detection in U2R and PROBING attacks.
7	M. MazharRathore, Anand Paul et al.	Real-Time Intrusion Detection for high speed networks	Hadoop based IDS for high speed real time intrusion detection. Nine best parameters are selected for intruder flows classification using FSR and BER, as well as by analysing the DARPA datasets[16].	REPTree and J48 algorithm	Proposed model has better efficiency and accuracy than existing models and is capable of handling big data.
8	MahsaBataghvaShahbaz et al.	Efficiency Enhancement of Feature Selection in IDS	Highly dimensional NSL-KDD dataset experimented on for feature extraction and selection for improving accuracy in IDS[15].	J48 classifier	Enhances performance through reduction of complexity and acceleration of detection process
9	FaridLawan Bello et al.	Analysis and Evaluation of Hybrid IDS	Different IDS classifier models analysed based on detection strategies calling for hybrid model to overcome limitations[14].	Support Vector Machine algorithm (SVM). Clustering based on Self Organizing Ant Colony Networks.	Hybrid model enables detection of multilevel classes of attacks with low classifier training time.

10	Ma Xiao-li et al.	Data mining in computer network security	KDD-CUP 2002 dataset to exploit to test out Artificial Immune System based classification for improved accuracy in intrusion detection.	Artificial Immune System algorithm; developed further on Neural Network and SVM classifier.	Accelerates speed of network intrusion detection. Reduces non-response rates and more reliable security model.
11	Subaira.A. S et al.	Improving Classification efficiency in IDS	Elucidates data mining as an efficient artifice for intrusion detection to determine key components from big data in networks [12].	SVM, decision tree Algorithms, Neural Network, , Bayesian Classifier, K- Nearest Neighbour, Fuzzy Logic and Genetic Algorithm	Reduces strain of physical compilations of the regular and irregular behaviour patterns.
12	Kailas Elekar et al.	Data mining in Intrusion Detection	Network traffic KDD – CUP dataset is scrutinized and supervised for detecting security faults using rule based data mining algorithm for detection[13].	Rule based data mining algorithm – OneR, PART, and zeroR, Decision Table, JRip.	Significantly better performance by PART classifier in overall intrusion detection classification.
13	Ali SharifiBoroujerdi et al.	DDoS Attack Detection	Ensemble of Sugeno kind adaptive neuro-fuzzy classifiers proposed for DDoS intrusion finding using Marliboost. Model performance evaluated on basis of detection of correctness and false positive alarms[9].	Fuzzy- Neural Network with Marliboost for boosting.	Proposed classifiers combination has improved detection accuracy to 96%.
14	Zeon Trevor Fernando et al.	Network Attacks Identification	Experimental analysis carried on KDD99 dataset and each feature is selected using integrated mechanism to identify attacks in the dataset[8][11].	J48 decision tree and Self-Organizing Map (SOM).	Increases overall classification accuracy by reducing dataset to prioritized subset.
15	ManasRanjanPatra and AshalataPanigrahi	Enhancing Performance of IDS	Soft computing techniques used on NSL-KDD dataset to assess performance of each procedure and determine most efficient solution for enhanced accuracy in intrusion detection[10].	Radial Basis Function Network (RBFN), Self-Organizing Map (SOM), Support Vector Machine (SVM), back propagation, and J48 classifier	Improved efficiency in cataloguing of network intrusion data into regular and irregular data.
16	V. K. Pachghare and ParagKulkarni	Pattern Based Network Security [13]	Highly uneven KDD-cup'99 dataset used as based to detect patterns using J48 graft for improved performance in intrusion detection[8].	J48 Graft algorithm (Decision Tree) and SVM classifier	J48 Graft tree determined to perform best for pattern classification in IDS.

4. Proposed Work

In the current k-NN algorithm the existing nodes are partitioned into classed and the result of applying the classifier is a membership to either of the classes.
k defines the number of neighbours in consideration.

When value of k=1, every training vector defines a section in space, defining a Voronoi partition of space.

$$R_i = \{p : \delta(p, p_i) < \delta(p, p_j) \mid i \neq j\} \tag{1}$$

Where, $R_i$is the radial distance of the neighbour i from the node.
Euclidean distance measure is used to calculate the distance between the node and its nearest neighbours.

$$\delta(x, y) = \delta(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2}$$

where δ is the distance between the nodes p and q.

Based on the distance vector, the k – instances are ranked by Bayesian probability, where the notations have their standard meanings,

$$P\,(M|E) = \frac{P(E|M)}{\sum_m P\,(E|M).P(M)} \,.P(M) \tag{3}$$

Alternatively, a sequential heuristic rank can also be assigned. The major drawback in this method of approach is the high search complexity that ensues Euclidean distance measurements. To overcome this drawback, only those computations that are absolutely necessary for getting an accurate measure should be computed. Thus, each node $n_i$is associated with an appropriate fractional weight w_i. The initial assignment of the weights is the same. Indicating equal initial weightage of all the nodes.

Furthermore, in order to limit the error in measurements, a constraint on the initial weight assignment has been imposed by stating that the sum of the weights of the kneighbours of an instance is 1.

$$\sum_{i=1}^{k} w_i = 1 \tag{4}$$

Each neighbour of n is assigned an initial value of 1/k. Based on the importance of each of the neighbours in determining the class of the new node, the weights are updated.Heuristic ranks assigned based on, probabilistic significance is used as a metric for weight updation.

$$w_i \longleftarrow w_i * h_i \tag{5}$$

The heuristic ranks can be determined by Maryam Kuhkan's updated measure of classification (from David Aha's model)[3][5],

Table 2. Weight Change Characteristics

Difference/Classification	Correct	Incorrect
Little	Unchanged	Much Decrease
Much	Unchanged	Little Decrease

By experimentally testing and trying out the method, it is found that the complexity could further be reduced by considering only the (k/2) + 1, most significant neighbours ranked in descending order by updated weights.

Thus, the new distance measure is,

$$\delta(x_p, x_q) = \sqrt{\sum_{i=1}^{\frac{k}{2}+1(sorted)} w_i * h_i * (x_{pi} - x_{qi})^2} \tag{6}$$

The, resultant vector is the list of distance measures of the node to its neighbours.Thus, the search complexity is significantly reduced.

5. Implementation

Weka 3.8 tool has been used for to implement the various lazy learning algorithms. NSL-KDD comprising of 22544 instances and 42 attributes was used as the representational dataset for real-world like network traffic data. Using 10-fold cross- validation testing option the classifier was deployed. A NetBeans framework was designed to incorporate the modifications of the novel distance vector measurements. The experimental implementation and observed results have been reported in the following tables.

6. Result

IBk –

Class wise accuracy –

Table 3. Performance Metrics of IBk

TP Rate	FP Rate	Precision	F-measure	ROC Area	PRC Area	Class
0.910	0.033	0.901	0.913	0.937	0.892	normal
0.936	0.049	0.945	0.933	0.937	0.953	anomaly
0.923	0.041	0.923	0.923	0.937	0.922	

Modified IBk –

Class wise accuracy –

Table 4. Performance Metrics of HW-IBk

TP Rate	FP Rate	Precision	F-measure	ROC Area	PRC Area	Class
0.969	0.019	0.974	0.972	0.976	0.962	normal
0.981	0.031	0.977	0.979	0.976	0.972	anomaly
0.976	0.026	0.976	0.976	0.976	0.967	

LWL –
Class wise accuracy –

Table 5. Performance Metrics of LWL

TP Rate	FP Rate	Precision	F-measure	ROC Area	PRC Area	Class
0.878	0.071	0.903	0.890	0.968	0.967	normal
0.929	0.122	0.910	0.919	0.968	0.973	anomaly
0.907	0.100	0.907	0.907	0.968	0.970	

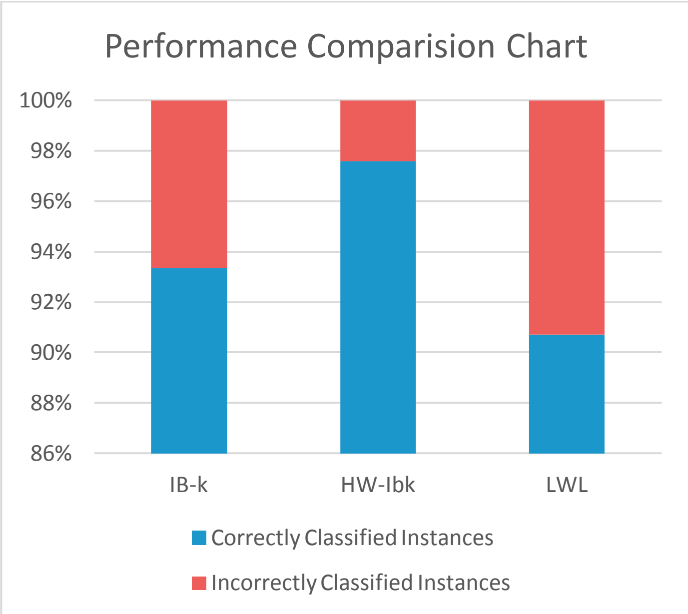

Fig1. Comparison of performance of lazy-learning classifiers on NSL-KDD

Thus, the overall accuracy has improved by nearly 4%, with reduced search complexity and thus computation results are available faster too. Not including less significant terms also prunes out noise introducing nodes during classification.

7. Conclusion

This paper elucidates the advantages of lazy learning in IDS. Lazy learning improves the efficiency of the NIDS by eliminating pre-fetching of overheads that are inherent in eager learning algorithms popularly in use today. Further an improvement of the k-nearest neighbour algorithm has been proposed to reduce the search complexity using a heuristic weight based indexing system. The results of this sufficiently prove thehw-IBk algorithm is a practical and viable solution for intrusion detection in data streams, with great accuracy,more so than other machine learning algorithms currently deployed. Additionally, the IBk algorithm has been compared to another other lazy learning algorithmLWL in order to compareand contrast their performances on the NSL-KDD network traffic dataset.The time taken to detect intrusions is significantly reduced and it is observed that the number of correctly classified instances of intrusions is relatively higher (~97.59).Thus, with significant increase in the speed of computation, network intrusions can now be detected faster without any loss to accuracy and thus aid in threat identification in real-time network system.

Acknowledgements

I would like to thank all the people who have motivated and helped me most throughout my project especially my colleagues who, by exchanging their own thoughts and providing valuable input made it possible to complete the paper with all accurate information.

References

[1] Lee, W., Stolfo, S.J. and Mok, K.W. (1999)"A data mining framework for building intrusion detection models." in Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium: 120–132.

[2] McHugh, J. (2000)"Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by Lincoln laboratory."*ACM Transactions on Information and System Security* (TISSEC), **3**(**4**):262–294.

[3] Noel, S. and Jajodia, S. (2005)"December. Understanding Complex Network Attack Graphs through Clustered Adjacency Matrices." in Proceedings of the 21st Annual Computer Security Applications Conference: 160–169.

[4] Zhang, J. and Zulkernine, M. (2006)"A Hybrid Network Intrusion Detection Technique Using Random Forests." in Proceedings of the First International Conference on Availability, Reliability and Security: 262–269.

[5] Maryam Kuhkan. (2006) "A Method to Improve accuracy of k-NN algorithm", IJCEIT**8**(**6**): 90–95.

[6]Cieslak, D. A., Chawla, N. V., &Striegel, A. (2006)"*Combating imbalance in network intrusion datasets*." in GrC: 732–737.

[7] Wang, W. and Battiti, R.(2006)"Identifying Intrusions in Computer Networks with Principal Component Analysis" in *Proceedings of the First International Conference on Availability, Reliability and Security* 270 –279.

[8] Pachghare, V.K. and Kulkarni, P.(2011)"Pattern based network security using decision trees and support vector machine." in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on* **5**(**1**): 254–257.

[9] Boroujerdi, A. S., &Ayat, S. (2013)"A robust ensemble of neuro-fuzzy classifiers for DDoS attack detection." in *Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference*:484–487.

[10] Patra, M. R., &Panigrahi, A. (2013)"Enhancing Performance of Intrusion Detection through Soft Computing Techniques." in *Computational and Business Intelligence (ISCBI), 2013 International Symposium:* 44-48.

[11] Fernando, Z. T., Thaseen, I. S., & Kumar, C. A. (2014)"Network attacks identification using consistency based feature selection and self-organizing maps." In *Networks & Soft Computing (ICNSC), 2014 First International Conference*:162–166.

[12] Subaira, A. S., &Anitha, P. (2014)"Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey." in *Intelligent Systems and Control (ISCO), 2014 IEEE 8th International Conference*:274–280.

[13]Elekar, K., Waghmare, M. M., &Priyadarshi, A. (2015)"Use of rule base data mining algorithm for intrusion detection." in *Pervasive Computing (ICPC), 2015 International Conference*:1–5.

[14]Bello, F. L., &Ravulakollu, K. (2015)"Analysis and evaluation of hybrid intrusion detection system models." in *Computers, Communications, and Systems (ICCCS), International* Conference: 93–97.

[15]Shahbaz, M. B., Wang, X., Behnad, A., &Samarabandu, J. (2016)"On efficiency enhancement of the correlation-based feature selection for intrusion detection systems." in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*: 1–7.

[16]Rathore, M.M., Paul, A., Ahmad, A., Rho, S., Imran, M. and Guizani, M.(2016)"Hadoop Based Real-Time Intrusion Detection for High-Speed Networks" in Global Communications Conference (GLOBECOM), 2016 IEEE: 1–6.