L-Semi-Supervised Clustering for Network Intrusion Detection

Srinivasa Rao Narisetty, Shaik Farzana, Potnuri Maheswari

ABSTRACT--- To identify and detect network intrusion attack is a challenging problem in the network communication. The major problem with these attacks is that they can exploit the network vulnerabilities and steal the sensitive information from the organizations. These intruders use polymorphic approaches to masquerade their identity to detect. In recent times, many supervised and unsupervised Machine Learning algorithms have been proposed to detect network attacks. Supervised learning requires labeled information to build a classifier. Indeed it requires do-main experts to label each attack. These issues are addressed by semi-supervised learning (SSL) approach where it builds a classifier from few labeled datasets. This paper proposes a novel leader based SSL approach by using labeled and unlabeled patterns to improve the performance of Intrusion Detection Systems (IDS). It has two step approaches- the first step it derives a set of prototypes by using a fast-clustering method along with constraints called the constrained leaders clustering method with threshold parameter ζ. The second step is by applying the single link method in the presence of a few labeled data with respective constraints. The experimental results are obtained from the standard dataset NSL-KDD which is an extension of KDDCUP-99 datasets where the proposed constrained leader-based SSL method achieved better accuracy even with few labeled training patterns.

Index Terms - Semi-supervised, Intrusion detection, Single link clustering, Machine Learning, Intrusion Prevention system

I. INTRODUCTION

Intrusion detection is the process of monitoring network traffic or host activities to identify the malicious intent that may cause damage to the machines connected on the network. Host based IDS (HIDS) monitors host specific events, generally using system log entries, and if it notices suspicious behavior it can identify the hosts who might have been compromised. Network based IDS (NIDS) detects activities that are suspicious with respect to network by examining network traffic. To detect and prevent (unauthorized) information leakage, network managers must deploy devices to prevent automatic distribution of information, while ensuring that the devices still allow network users to leverage the benefits of network resources effectively. IDSs generally use two techniques for detecting intrusions: 1. Anomaly-based IDS and 2. Signature based IDS. Anomaly-based IDS use a representation of common activities on the network and look for deviations from common activities.

Based on heuristics, if the deviations show a pattern close to what might be considered malicious, a suspicious event-alert is raised. NIDS monitor the network traffic continuously and compare the packets against known attack patterns. The collection of these patterns is generally called signatures/rules. The usage of these signatures accurately identifies known attacks, but is ineffective for anonymous attack patterns. On the other hand, anomaly-based IDS detect anonymous attack patterns using several techniques viz. statistical based, knowledge based and machine learning based [3] techniques. The main issue with anomaly-based detection is that false positive (FP), and false negative (FN) alarms [9] are large in numbers. The growth of wireless connectivity is also opening another challenge. Enterprises can no more feel secure by controlling the traffic on the wired WAN interface. With the need for IDS/Intrusion Prevention System (IPS) to monitor Intranet traffic, efficiency of these solutions has to be optimized.

The detailed analysis of KDDCUP-99 is discussed in [14] to detect network intrusions based on the behavioral patterns. To increase the efficiency of IDS/IPS various approaches have been applied viz., supervised and unsupervised approaches. In supervised approach, the training data is labeled i.e., every pattern indicates the class either ``normal'' or ``anomaly''. Many supervised algorithms viz., Decision-trees (C4.5), Support Vector Machines (SVM), Neural Network (NN) and k-Nearest Neighbor (KNN) are used by research community to accurately detect in real-time. This method is very accurate when the dataset has labels. On the other hand, unsupervised learning is most popular to group the similar patterns. It learns the dataset with unlabeled instances. Clustering is the most popular unsupervised approach that finds the similarities among the patterns to build clusters. The intuition behind this approach is the same cluster patterns that having similar properties and categorized into same class. This results low accuracy when compared to supervised approach. However, it has better prediction rate than supervised approach which is more robust for IDS/IPS.

In [5] the classification is live, though using data generated in a lab environment along with replay of collected data. They have a limitation on the generation of network traffic (eight applications used) and attacks were limited to port scan, denial of service and UDP flooding (fraggle). A total of around 500 traffic sessions were created and attack traffic was generated for an average duration of five minutes.

Revised Manuscript Received on December 22, 2018.

Srinivasa Rao Narisetty, Lakireddy Bali Reddy College of Engineering, Mylavaram, Krishna-Dist, Andhra Pradesh, India – 521230. (E-mail: nsr543@gmail.com)

Shaik Farzana, Lakireddy Bali Reddy College of Engineering, Mylavaram, Krishna-Dist, Andhra Pradesh, India – 521230. (E-mail: shaik.farzana52@gmail.com)

Potnuri Maheswari, Lakireddy Bali Reddy College of Engineering, Mylavaram, Krishna-Dist, Andhra Pradesh, India – 521230.(E-mail: potnuri.maheswari@gmail.com)

In [6] live traffic collected from Beta Site was used which generates around 100 Gbytes/hr. By using two-tier architecture to capture this traffic (called PCAPLib system) was used effectively to capture traffic while maintaining anonymity. These captures were later played back within the benchmark system. The novelty of the data collection while preserving anonymity and discarding irrelevant information is uniqueness of this approach.

The traditional single-link clustering method, can find arbitrary shaped clusters [7] where k-means clustering method can't find. The limitation of ``k-means'' is that can find only spherical shaped clusters. The single-link method comes under the categorization of hierarchical clustering methods where it follows bottom up approach. The final result is evolved based on the user's selection criteria. The Semi-supervised single-link method is having two demerits, i.e. (i) The computational cost of the existing SSL method and (ii) the final result is chosen from user's criteria. Constrained leader-based SSL method can overcome the above two problems, and also produces the almost similar clustering result and much reduced time.

This work proposed an unsupervised approach and leader based semi-supervised learning (SSL) approach which would be the best choice to identify/detect the new attacks that can improve the accuracy of IDS/IPS. This paper proposes a method called constrained leader-based semi-supervised single-link clustering method which is a two-step process where prototypes are generated in the first step with the help of fast clustering method called leaders method with incorporation of instance level constraints. The method, purely based on leaders algorithm, first derives a set of leaders set and associated followers come under specific leader, based on threshold value ζ and also labels are considered into account.

The remaining paper is arranged as follows. In Section II reviews about semi-supervised learning (SSL) method. Section III explains proposed leader-based SSL using single link clustering. The experimental evaluation is presented in Section IV along with results. Discussions on experimental results, comparative analysis and conclusions are presented at the end.

II. SEMI SUPERVISED CLUSTERING

Semi supervised clustering is a special case of clustering. Generally in clustering we use unlabeled data patterns for clustering. But in semi supervised clustering we use both labeled and unlabeled data with side information as pair wise(must-link and cannot link) constraints which helps to cluster the data patterns [15].Self-training is most commonly used technique in SSL. In self-training, a classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data, the predicted labeled data patterns added to the training set. The process is repeated until the test set get empty, where the classifier uses it own predictions to teach itself. This process is called self-teaching or bootstrapping. Few algorithms try to avoid by 'unlearn' unlabeled points if the predicted data patterns are below threshold. Self-training has been applied in many applications such as natural language processing tasks [4]. Authors [12] pro-posed semi-supervised self-training method to detect object detection from images n computer vision application.

[11] apply self-training in face recognition system which improve the performance. However, self-training approachis difficult to analyze in general.

III. PROPOSED LEADER BASED SEMI-SUPERVISED SINGLE-LINK APPROACH FOR NIDS

In this section, we discuss proposed method to detect network intrusion detection. The framework of this work is proposed in Fig. 1. The framework has several phases:

Algorithm 1: Constrained Leaders set

Algorithm Leaders(X, ζ)
let us assume the first pattern is always a leader and has label
Let $\mathcal{L}$ be the leaders set
Let $\mathcal{F}$ be the followers set
let $\mathcal{L} \leftarrow$ x
let $\mathcal{F} \leftarrow \phi$
for $x \in X$ **do**
  Find a $l \in L$ such that distance $\|l - x\| \leq \zeta$ and x is labeled
  If there is no such l then
  if *label(l)* $\neq$ *label(x)* **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{x\}$;
    count$(x) \leftarrow 1$;
    count$(l) \leftarrow$ count$(x) + 1$;
    if *label(l)* $=$ *label(x)* **then**
      $\mathcal{F} \leftarrow \mathcal{F} \cup \{x\}$;
    else
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{x\}$;
      count$(x) \leftarrow 1$
  else
    If the leader l is unlabeled such that distance$(x, l) \leq \zeta$ & x is labeled
    label$(l) =$ label(x);
nfleaders $=$ len$(\mathcal{L})$;
return <$\mathcal{L}$, $\mathcal{F}$, #nfleaders> ;

i. **Pre-process Phase**: In this phase, the features are standardized irrespective of data is labeled or unlabeled.
ii. **Selection of patterns**: In this phase, proposed leaders algorithm applied to select important patterns to reduce the running time of overall training and testing phase.
iii. **Training Phase**: During this phase, the selected patterns are partitioned as labeled and unlabeled to detect the intrusions using single-link clustering algorithm.
iv. **Testing Phase**: The unseen patters are supplied to training model to predict normal and anomaly intrusion.

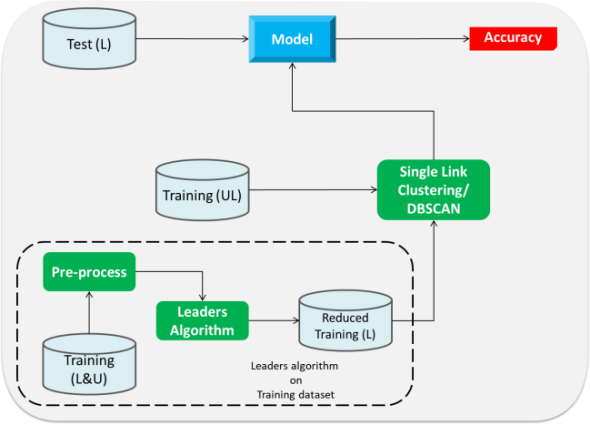

Fig 1: System Model

The proposed leader based pattern selection is shown in Algorithm 1. Given any pat-terns, 1 finds the leader set and followed by followers set. Every leader may have follower(s) set that indicates redundant pattern. The proposed approach only utilizes the leaders set and classifies.

A. Leader based Semi-supervised algorithm for NID

Our objective is to reduce the number of patterns in the training set that impact the computational time of Single-link to for the clusters using leaders' mechanism. The leader based Semi-supervised algorithms, has a training data with very small samples of labels and a tuning parameter ζ. At the beginning, the ζ is derived from the dataset and which is fixed throughout the algorithm. The intuition behind leaders Algorithm 1 is that first pattern is always a leader and finds the followers subsequently. The leaders are formed such that distance from the current pattern and the next pattern more than ζ, otherwise the pattern is treated as follower of the current pattern. At the same time, the algorithm also has constraint that the distinguished label patterns cannot be at the same group. Algorithm 1 determines the leader patterns and number of leaders count with matching labels.

The *L* be the leaders set has the first pattern treated a leader and *F* be the followers set is initialized as empty set. In our proposed method the ζ has interval from [0.0-0.2-2.0] for all values.

B. Description of KDD Network Intrusion Detection dataset

The original KDD-CUP99 is modified and derived a new dataset to identify intrusion activity over a network. The drawback of KDD 99 [8] has proven statically and measured the accuracy of IDS/IPS by research community. The NSL-KDD dataset. The researchers from University of New Brunswick (UNB) are measured the following improvements from KDD'99:

1. The train and test dataset don't have duplicated records, so that the classifiers will not bias towards the more frequent patterns.
2. The choice of the records is inversely proportional to the KDD'99 from each difficulty level group that result, the classification rate of distinct machine learning algorithms vary in a wide range.
3. The generated newer version dataset is computationally affordable. The results of different research works are consistent and comparable.

There are 41 features are extracted from every flow and labels assigned to each either normal or an attack type. The features are categorized into 6 groups viz., basic features, domain knowledge, same hosts, same services, destination hosts and time based features. The NSL-KDD dataset has many continuous and very few discrete type of features. The original dataset has 4 different attack types are classified namely: DOS, Probe, R2L and U2R.

The description of the 4 attack types are defined below:

i. **Denial of Service (DoS)**: Denial of service attack exhaustive the victim's system resources that unable to handle legitimate requests to and fro. Example attacks are: Syn, ICMP, UDP and Ping of Death etc.
ii. **Probing (Prob):** Probing attack, where the attacker may scan the victim's system or a network to exploit the vulnerabilities that may lead to compromise the system. Example attacks are: saint, port sweep, mscan, nmap and Xscan etc.
iii. **Remote to User (R2L):** R2L attack attempts an unauthorized access from remote to local victim's machine that gain an access. Example attack can be password guessing.
iv. **User to remote (U2R):** U2R attack attempts to unauthorized access to root user i.e. super user, by which an attacker login as normal user into victim's machine to gain an access. Example attacks are: buffer overflow attack.

IV. EXPERIMENT RESULTS AND DISCUSSIONS

This section describes the detailed analysis of experimental analysis to identify intrusion behavior as abnormal activity. We have two stages in our experiments, in the first stage the leader patterns are derived from Algorithm 1. In this stage, there are two set of patterns are derived called leaders and followers based on threshold ζ. The leaders indicates that these patterns are adequate to represent entire dataset. Similarly, the follower patterns indicates that these patterns are redundant and are not part of in the training phase. The leaders approach reduce learning task time. Once the leader patterns are filtered, then Single-link clustering is performed to measure the performance.

The training dataset has patterns with 10% of labels and 90% of them unlabeled. There are 10 different training datasets are generated using leaders algorithm by changing the input parameter ζ. The Single-link method tray to group them as normal and anomaly based on minimum distance criteria as Euclidean measure. Once the clusters are formed, the testing accuracy evaluated. The model build with single-link clustering using KDDTrain+_20Percent dataset and the test clustering accuracy obtained on both KDDTest+ and KDDTest_21 dataset. Table 1 describes the accuracy of various subset of training datasets and performance is measure on test datasets respectively.

Table 1: Accuracy of NSL-KDD dataset based on parameter ζ

ζ	KDDTrain+_20 Percent	KDDTest+	KDD Test_21	Running Time	Training set size
0.5	0.93	0.78	0.72	0.52	23119
1	0.87	0.75	0.69	0.48	20500
1.5	0.89	0.71	0.65	0.3	18567
2	0.87	0.68	0.5	0.29	16098
2.5	0.86	0.6	0.48	0.22	14200
3	0.86	0.58	0.47	0.09	11067
4	0.81	0.56	0.48	0.08	10671
4.5	0.81	0.53	0.43	0.07	9067
5	0.78	0.51	0.42	0.05	7019

We notices that as the ζ changing increasingly the computational time is reduced on training set. The running time is measured only for training phase not for testing phase. With complete KDDTrain+20Percent dataset, the highest accuracy is observed 0.95 and further as the tuning parameter increasing and proportionally the detection rate is also reduced. It is clearly shown that the impact of leader based single clustering has trade-off between accuracy and computational time. If there is a reduction in training dataset then reduction in accuracy but the computational time is minimized. With ζ 0.2, the training accuracy is 0.93 and testing accuracy for both is 0.78 and 0.72 respectively, which very slight deviation in the detection rate but greatly reduce the learning time. The number of training patterns are even further reduced based on tuning parameter ζ. With ζ = 0.1 the training sample reduce from 25192 to 24011. The10-fold accuracy is yielded to 0.95 and the prediction on both the testes is 0.82and 0.74 respectively.

Table 2: Describes the percentage of labeled dataset and Accuracy on NSL-KDD

Percentage of labels	KDDTrain+	KDDTest+	KDDTest-21
1	0.7	0.52	0.4
2	0.72	0.55	0.44
3	0.77	0.58	0.48
4	0.8	0.62	0.49
5	0.88	0.65	0.54
10	0.9	0.69	0.59
20	0.92	0.7	0.61
30	0.95	0.73	0.67
40	0.95	0.75	0.71
50	0.97	0.8	0.77

We also study the impact the of labeled training dataset with same leader set on large number of records is shown in Table 2. The KDDTrain+ dataset has125973 patterns are found and class labels are assigned. On the other hand theKDDTrain20Percent has 25192 patterns is about 20% of KDDTrain+. Without change in ζ there are 10 different subset of training dataset are generated using Algorithm 1. The other experiment, we vary the percentage of labeled dataset from1%–50% to evaluate the performance of single-link with leaders. We observed, as the labeled training dataset increases linearly the accuracy also increased linearly on train and test datasets. The lowest accuracy of 10-fold cross-validation of KDDTRain+ is 0.70 with 1% of labels and the highest accuracy 0.97 with only 40% of labels. Similarly, the detection rate on test dataset also increased linearly.

The experimental results are demonstrated in Fig. 2 using leader based semi-supervised single-link clustering approach. Fig. 2a demonstrates the accuracy of KDD training of Single-link clustering with 10-fold cross-validation and for both test datasets.

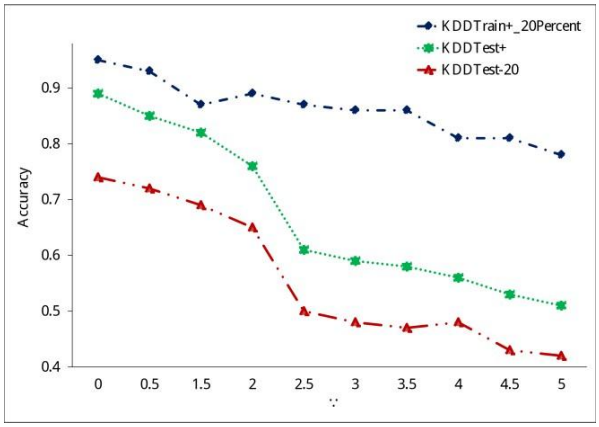

Fig (a): KDDTrain20Per of 10-fold, KD-DTest+ and KDDTest-20

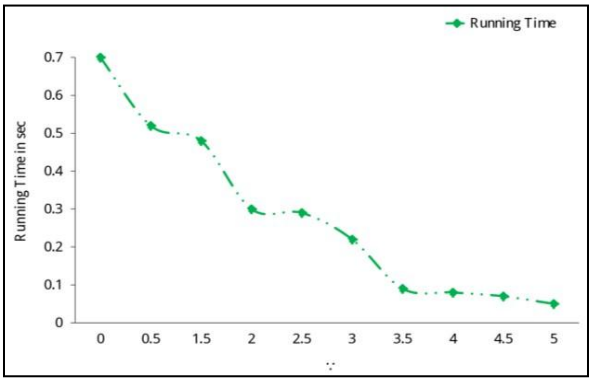

Fig (b): Running Time of Training model

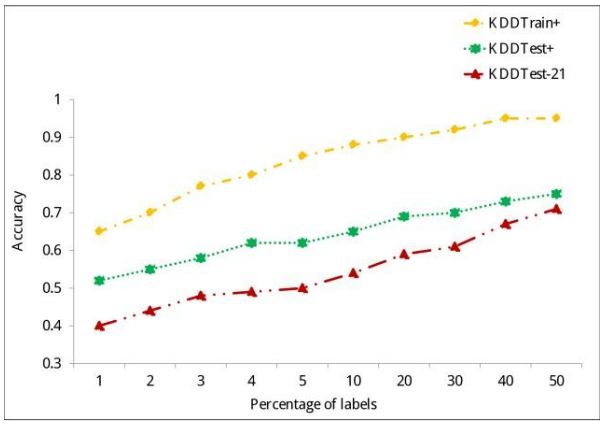

Fig (c): KDDTrain+ dataset for 10-fold, KDDTest+ and KDDTest-20 by varying percentage of label data

Fig. 2: Performance of l−SSL Single-Link on KDD Network Intrusion Detectionw.r.t ζ

In our proposed method the accuracy of KDDTest+ and KDDTest21 dataset observed to be 0.82, 0.74 respectively. Hence, it is verified that the leader based semi-supervised single-link can enhance the performance of IDS/IPS to detect intrusion in real-time.

V. CONCLUSION

In this paper we have developed a new approach leader based SSL for improving the classifier performance on IDS/IPS by introducing leaders concept that can work with less training examples with labeled and unlabeled records with Single-Link clustering approach because it is very efficient and utilizes very limited resources. The experiments demonstrated in this paper is reduced the classification accuracy by introducing leaders concept with clustering mechanism that lead to misclassification rate. It is noticed that our methodology is an effective way to improve the classification accuracy even with less number of training patterns with partial labeled samples. The classifier is performed better when incorporating the unlabeled patterns with their predicted labels into the original training set. In this paper, we reported two-class problem, i.e., normal and anomaly. Further, our research work will continue towards applying more semi-supervised clustering approaches to improve the effectiveness of IDS/IPS for detecting multiple types of attacks.

ACKNOWLEDGEMENTS

Authors are thankful to the “Research Center” recognized by the JNTU, Kaki-nada and Krishna University of Computer Science and Engineering, LBRCE for providing infrastructure facilities during the progress of work.

REFERENCES

1. R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He. Fuzziness basedsemi-supervised learning approach for intrusion detection system. Information Sciences,378:484–497, 2017.
2. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the eleventh annual conference on Computational learning theory, pages92–100. ACM, 1998.
3. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maci a-Fern ande and E. a que . Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security,28(1-2):18–28, 2009.
4. G. R. Haffari and A. Sarkar. Analysis of semi-supervised learning with the yarowsky algorithm. arXiv preprint arXiv:1206.5240, 2012.
5. I. U. Haq, S. Ali, H. Khan, and S. A. Khayam. What is the impact of p2p traffic on anomaly detection? In International Workshop on Recent Advances in Intrusion Detection, pages1–17. Springer, 2010.
6. C.-Y. Ho, Y.-C. Lai, I.-W. Chen, F.-Y. Wang, and W.-H. Tai. Statistical analysis of falsepositives and false negatives from real traffic with intrusion detection/prevention systems.IEEE Communications Magazine, 50(3), 2012.
7. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM computing surveys (CSUR), 31(3):264–323, 1999.
8. S. S. Kaushik and P. Deshmukh. Detection of attacks in an intrusion detection sys-tem. International Journal of Computer Science and Information Technologies (IJCSIT),2(3):982–986, 2011.
9. S. Kumar and R. Joshi. Design and implementation of ids using snort, entropy and alert ranking system. In Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, pages 264–268. IEEE, 2011.
10. F. Massicotte and Y. Labiche. An analysis of signature overlaps in intrusion detection systems. In Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on, pages 109–120. IEEE, 2011.
11. F. Roli and G. L. Marcialis. Semi-supervised pca-based face recognition using self-training. In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pages 560–568.Springer, 2006.
12. C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models, 2005.
13. M. Seeger. Learning with labeled and unlabeled data, 2000.
14. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup99 data set. In Computational Intelligence for Security and Defense Applications, 2009.CISDA 2009. IEEE Symposium on, pages 1–6. IEEE, 2009.
15. [15].X. Zhu. Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison, 2(3):4, 2006.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication