Pragmatic Analysis of Machine Learning Techniques in Network Based IDS

Divya Nehra(✉), Krishan Kumar, and Veenu Mangat

University Institute of Engineering and Technology,
Panjab University, Chandigarh, India
divyanehra@gmail.com, k.salujauiet@gmail.com,
veenumangat@yahoo.com

Abstract. In providing defense to computer networks the network intrusion detection system (NIDS) plays a very essential role. To cope up with the demands of contemporary networks various concerns like performance evaluation and others related to the networks should be taken under consideration. Proposed work presents a pragmatic analysis of machine learning techniques for network based IDS. The performance analysis over two benchmark datasets i.e. KDD-Cup'99 and NSL-KDD by using five supervised machine learning techniques (RFC, Naïve bayes, J48, Bayes Net and SVM) has been prepared. To assess the performance network based intrusion detection system various metrics such as accuracy, recall, F1-score and precision has been computed and analyzed. Therefore, the summary of the work suggests that no single technique is smart enough to identify all attack classes to conventional levels. Most of the techniques provided poor results for minority attack class(es). To estimate and assess the supervised classifier a blind set of investigation with 10-fold cross validation has been performed. The results achieved are promising and provides a new direction to researchers of the intrusion detection domain.

Keywords: Network security · Intrusion detection system · Security performance · Network intrusion detection · Security and protection

1 Introduction

Network based IDS are the software employed within the networks at some deliberated point to analyze network circulation on the whole subnet. The traffic log is matched through the database of recognized attacks and if a spasm is spotted or security policy violation is detected, a signal is passed to the network supervisor. NIDS are classified as On-line NIDS and Off-line NIDS. On-line NIDS are those which are able to work with the real-time networks whereas the Off-line ones are those who works over the repository of data and analyze the data in such a way to identify the attacks and normal instance.

In recent trend, the main attention of researchers has been inclined towards the machine learning techniques and neural network techniques like Random Forest, Support Vector Machines, Naïve Bayes and Decision Trees [1]. These techniques have been achieving better and improved performance in detection accuracy for network1

A. K. Luhach et al. (Eds.): ICAICR 2019, CCIS 1075, pp. 422–430, 2019.
https://doi.org/10.1007/978-981-15-0108-1_39

intrusion detection system. Machine learning taxonomy has given a whole new meaning to the field of intrusion detection when used up to its potential [2, 3].

To address the improvement required in the field of intrusion detection this new strategy is proposed.

2 Background

This section provides the related material which is necessary to realize the stimuluses and the idea behind the anticipated work in this work.

2.1 Network Intrusion Detection System

Now a day dependence over the organizations that relies on gradually demanding application of information technology is increasing rapidly. Thus service provider software is more prone to vulnerabilities and the errors involved are economically high in cost to be solved. This scenario leads to the need and innovation of a strong network monitoring system which can deal with the following pertinent concerns:

- **Dimensionality of data:** The dimensionality of stored as well as passing by data over network is increasing massively and will be continue to increase. According to the forecast made in [4] the amount of data will reach up to 44ZB by 2020. Deploying NIDS to deal with such big amount of data is a major challenge.
- **Reliability:** To achieve desired levels of reliability in terms of accuracy, the existing techniques are somewhere lacking. Hence more granular datasets, more visualization of data is required to achieve more promising results.
- **Mélange:** The present scenario is focusing on developing ensemble and customized protocols using various algorithms and network attributes. Consequently, identification of nefarious and normal behavior is becoming a cumbersome task.
- **Imbalanced datasets:** This problem arises when datasets consist of such classes which has fewer or smaller number of instances. Due to it, NIDS becomes unable to precisely predict such classes and becomes more prone to errors.

2.2 Machine Learning

According to Wikipedia, machine learning is subclass of artificial intelligence in the domain of computer science that empowers the computers with the ability to “learn” the data by using the statistical techniques, without being explicitly programmed [5]. Therefore, machine learning is programming the computers to enhance a benchmark efficiency via past practice or stored data. Machine learning make uses of the philosophy of statistics to build up mathematical prototypes to make out a corollary from an illustration. Various example of machine learning applications is basket analysis using learning associations which says 70% of customers who buy bread also buy butter, classification problem in which two or more classes are present and by making use of machine learning algorithms the appropriate class of the instance is predicted, pattern

recognition which consists of face recognition, medical diagnosis and speech recognition etc. [6]. Machine learning algorithms are divided as following types:

- Supervised learning: the aim of this learning is to memorize the patterns or mapping of input to output whose labels or results are provided by the supervisor himself [7].
- Unsupervised learning: in this type of learning no supervisor is present and only input is provided. Here, the goal is to discover the symmetries in the input. The concept of clustering is used here to make clusters of similar patterns [8].
- Reinforcement learning: this learning selects an action out of sequence of actions and learns the policy which was being used by the sequence of actions to reach the goal. Here the aim is to learn the goodness of policies and generate a policy [6, 9].

3 Existing Work

In this section, the most recent prominent works has been discussed.

The goal NIDS using machine learning is to breed a minimal rule set to detect malicious actions deviating from past behaviors. There are quite a few existing workings in the field of Network IDS. The work by [8] propose a new method to Network intrusion detection and achieved a FNR = 1.15%, FPR = 0.09% and detection accuracy of 98.76% in comparison to another SVM based scheme they've achieved FPR = 4.2%, FNR = 7.77% and detection accuracy of 88.03%. [10] propose a machine of generating learning model for NIDS by comparing five machine learning based models and achieved detection accuracy of 99.4%. They've compared the results with reduced feature set and without reduced feature set. Moreover, one more comparison is made between 10 fold cross-validation results and percentage split results.

[11] propose a machine learning based approach using SVM with augmented features. They have implemented the marginal density ratios transformation method to obtain improved detection rate for SVM. The dataset used is NSL-KDD and the results shows the robust performance results. [12] proposes an IDS on the basis of performance comparison between SVM, RFC and ELM to resolve concerns of performance. The use of these techniques shows limitations of large datasets, huge traffic data and gives an efficient classification technique. [13] analyzed methods for management of datasets related to imbalacing and they concludes that minority classes are not capable for learning as compare to majority classes. [14] has discussed problems regarding learning with skewed class scatterings and effect of it over performance of classifiers. The analysis was conducted for artificial intelligence and computational intelligence and confirms the requirement of building efficient intrusion detection systems. In [23], analysis of artificial NN, decision tree, support vector machine, Bayesian networks and a self-organizing map has been done. Even though high and desirable results have been achieved using machine learning but still machine learning consists of some vulnerabilities, such as misclassification of network data due to poison learning. Such vulnerabilities in the system affect performance. So such problems of machine learning need to be addressed.

4 Classifier Used

In our proposed work, following five algorithms have been used on two different datasets i.e. KDD Cup'99 and NSL-KDD. 10-fold CV approach has been applied with the help of Scikit Learn.

- Random Forest Classifier: these classifiers are from the family of ensemble or forest of decision trees. This family generally have low bias and high variance and are perfect contenders for ensemble method. The bootstrap aggregating or bagging technique is generally used in this classifier to achieve increased variance without altering the bias [15].
- J48: it is a predictive learning technique which make predictions for the new instance on the basis of prior available information. It creates a decision tree using the values of available data [16].
- Naïve Bayes: these classifiers belongs to the family of probabilistic classifier. It uses bayes rule of conditional probability. Naïve bayes observes each feature individually as well independently of other features contained by model [17].
- SVM: these classifiers are best suited for multiclass classification problems for big datasets and one of the superfast machine learning classifier with low computational resources [18]. This family supports classification as well as regression.
- BayesNet: These are the sub set of Bayesian networks with nominal attributes and no missing values [19].

5 Calculations

Related to most of the existing research, our proposed work was implemented using Python. All evaluation was performed using 64-bit Windows 10 Pro with an Intel® Core™ i5-8250 CPU @ 1.60 GHz 1.80 GHz with 8.00 GB RAM and an NVIDIA GeForce MX150 GPU. Two of the benchmark datasets of the domain of intrusion detection i.e. KDD Cup'99 as well as NSL-KDD datasets are used for performance evaluation.

The used metrics are as follows:

- True Positive(TP) – those occurrences which are correctly categorized as an intrusion.
- False Positive(FP) – those occurrences which are incorrectly categorized as an intrusion.
- True Negative(TN) – those occurrences which are correctly categorized as normal.
- False Negative(FN) – those occurrences which are incorrectly categorized as normal.

Performance of the proposed work is calculated by using the following measures:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The measure of accuracy is appropriately identified instances to the total number of records.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The precision is the measure of correctly identified records to the incorrectly identified records.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The recall is the measure of correctly identified records to the number of missed records.

$$\text{F1} - \text{Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

The F1-Score is the measure of harmonic mean of recall and precision.

5.1 Datasets

Two datasets have been used i.e. NSL-KDD and KDD-Cup'99. They are publically available benchmark datasets and have been massively used by the researchers of intrusion detection domain.

KDD Cup'99: In 1998 MIT Lincoln Labs prepared the intrusion detection assessment program named as DARPA IDS evaluation program. The network log consisting of intrusions imitated in military network environment for survey purpose was conducted [20]. Later on, the KDD Cup'99 dataset utilized it. This dataset contains 4900000 number of records with 41 type of features (e.g. duration, flag, land) and these features are broadly classified into three main classes. As it is a labelled dataset so each record is labelled as normal or attack (attack type). Most of the researchers make use 10% subset of original dataset as working with it requires less computation. The dataset needs to be pre-processed before usage. The pre-processing consists of transformation of string or symbolic values to numeric values to make learning easier.

NSL-KDD: The NSL-KDD is the improvement over KDD Cup'99 with reduced number of redundancy. The number of features is same as of KDD Cup'99 [20], [21]. Though this dataset has also faced criticism but still it is being used extensively world-wide. Whole of the dataset has been used for 5-class classification. Following are the various reason to use NSL-KDD (Table 1):

1. Redundant records are not present in train dataset so classifier is free from producing biased results.
2. Test dataset is free from duplicate records which helps in better reduction rates.

6 Results and Discussions

Results obtained are indicating that out of all the classifier used, RFC is performing the best in terms of Accuracy, Precision, Recall and F1-Score. Moreover, one more analysis is made regarding the number of records available for the R2l and U2r class are less as compare to other classes so is the accuracy and other metrics is also low.

Table 1. NSL-KDD 5-Class Performance

Classifier	Metrics	DoS	NORMAL	PROBE	R2L	U2R
RFC	Accuracy (%)	91.59	99.29	98.45	93.56	15.5
	F1-score (%)	92.00	99.04	99.78	97.00	19.57
	Precision (%)	99.54	99.79	99.89	97.89	66.99
	Recall (%)	91.59	99.29	98.45	94.56	15.5
J48	Accuracy (%)	96.32	93.54	76.44	6.9	15.86
	F1-score (%)	97.00	94.54	77.10	10.54	19.42
	Precision (%)	99.00	99.00	99.00	97.00	66.99
	Recall (%)	96.32	93.54	76.44	6.9	15.86
BayesNet	Accuracy (%)	94.5	97.5	83.45	55.75	14.78
	F1-score (%)	95.00	98.07	84.9	55.95	19.33
	Precision (%)	99.00	99.00	97.12	97.45	65.50
	Recall (%)	95.99	97.99	84.97	55.97	15.25
Naïve Bayes	Accuracy (%)	84.3	96.50	78.56	57.44	19.44
	F1-score (%)	85.4	96.99	78.99	57.95	21.50
	Precision (%)	99.41	99.74	99.00	93.00	41.41
	Recall (%)	84.3	96.50	78.56	57.44	19.44
SVM	Accuracy (%)	90.49	94.71	96.39	83.71	13.59
	F1-score (%)	91.48	94.94	97.83	94.17	14.00
	Precision (%)	99.00	94.14	97.78	93.11	15.05
	Recall (%)	91.05	95.02	97.99	94.41	14.00
Total Instances		45927	67343	11656	995	52

6.1 KDD Cup'99 Evaluation

This section provides the evaluations made on KDD Cup'99 dataset.

5-Class Classification: 5-Class classification consists of the standard 5 classes i.e. Normal, DoS, U2r, Probe, R2l. 10% subset of KDD Cup'99, which is a common practice, has been used. The results indicate that 2 out of 5 classes shows poor

performance i.e. R2l and U2r. The rest of the classes offer significant level of accuracy, precision, recall and f1-score. Moreover, it can also be observed from the results that the overall performance of Random Forest Classifier is the best and SVM also outperforms whereas naïve bayes is the worst performer in terms of accuracy (Table 2).

Table 2. KDD-Cup'99 5-Class Performance

Classifier	Metrics	DoS	NORMAL	PROBE	R2L	U2R
RFC	Accuracy (%)	95.34	98.97	96.96	81.95	12.9
	F1-score (%)	96.11	97.99	96.99	81.95	14.75
	Precision (%)	98.98	97.98	96.96	82.94	12.66
	Recall (%)	95.34	98.97	96.99	81.96	12.9
J48	Accuracy (%)	61.96	96.97	95.96	60.85	14.73
	F1-score (%)	62.96	97.96	96.98	61.94	15.75
	Precision (%)	66.96	95.95	96.96	64.95	17.86
	Recall (%)	61.95	96.96	95.99	60.96	14.67
BayesNet	Accuracy (%)	87.97	76.97	68.87	55.19	13.23
	F1-score (%)	88.98	77.97	69.98	56.96	14.25
	Precision (%)	90.98	80.97	70.96	60.93	10.86
	Recall (%)	87.97	76.96	68.99	55.96	13.67
Naïve Bayes	Accuracy (%)	56.60	90.95	70.76	50.53	15.54
	F1-score (%)	57.95	91.97	70.97	50.95	10.66
	Precision (%)	59.96	90.96	70.95	50.94	10.78
	Recall (%)	56.96	90.97	70.98	50.96	15.77
SVM	Accuracy (%)	94.93	92.95	80.94	70.26	14.17
	F1-score (%)	94.92	93.95	80.97	70.95	13.86
	Precision (%)	95.97	90.96	81.96	70.97	13.85
	Recall (%)	94.97	92.96	80.94	70.54	14.00
Total Instances		391458	97278	4107	1126	52

7 Conclusion and Future Work

This work has used the benchmark datasets KDD Cup'99 and NSL-KDD to make performance evaluations. The comparisons have made between 5-class classification of both the datasets. On comparison, we found that the RFC is performing the best in both scenarios. Moreover, it may also be noted that the classes like U2r and R2l are not giving very promising results because of the number of instances available for training. It suggests that efforts for refining the performance of present techniques for rare attack classes needs instant addressing by scholars. Moreover, the results obtained also suggests that for a particular attack class, some classifiers perform better than the others. The significant reason for that is different algorithms are designed differently to work with their particular characteristics.

In future work, the improvement will be made in the direction of dealing with class imbalancing problem. We will work upon improvement of existing evaluations by utilizing more efficient methods like shallow learning and deep learning. Hence we can extend the proposed work to achieve more and more merits out of it.

References

1. Dong, B., Wang, X.: Comparison deep learning method to traditional methods using for network intrusion detection. In: 8th IEEE International Conference Communication Software Networks, pp. 581–585 (2016)
2. Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Tech. Rep. **99**, 1–15 (2000)
3. Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., Atkinson, R.: Shallow and deep networks intrusion detection system: a taxonomy and survey. CoRR, abs/1701.0, pp. 1–43 (2017)
4. Executive summary: Data growth, business opportunities, and the IT imperatives—The digital universe of opportunities: Rich data and the increasing value of the Internet of Things. https://www.emc.com/%0Aleadership/digital-universe/2014iview/executive-summary.htm
5. Machine learning. https://en.wikipedia.org/wiki/Machine_learning
6. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2010)
7. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: Proceedings of the IEEE Symposium Security Private, pp. 305–316 (2010)
8. Chowdhury, M.N., Ferens, K., Ferens, M.: Network intrusion detection using machine learning, pp. 30–35 (2010)
9. Alpaydın, E.: Introduction to machine learning. Methods Mol. Biol. **1107**, 105–128 (2014)
10. Kumar, S., Viinikainen, A., Hamalainen, T.: Machine learning classification model for network based intrusion detection system. In: 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 242–249 (2016)
11. Wang, H., Gu, J., Wang, S.: An effective intrusion detection framework based on SVM with feature augmentation. Knowl.-Based Syst. **136**, 130–139 (2017)
12. Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A.: Performance comparison of support vector machine random forest and extreme learning machine for intrusion detection. IEEE Access **6**, 33789–33795 (2018)
13. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Handling imbalanced datasets: a review. In: GESTS International Conference on Computer Science and Engineering, vol. 30, pp. 25–36 (2006)
14. Monard, M.C., Batista, G.E.A.P.A.: Learning with skewed class distribution. In: Advances in Logic, Artificial Intelligence and Robotics, Sao Paulo, SP, pp. 173–180. IOS Press (2002)
15. Random Forest Classifier. https://www.stat.berkeley.edu/∼breiman/randomforest2001.pdf. Accessed 19 April 2018
16. Sahu, S.: Network intrusion detection system using J48 decision tree. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2023–2026 (2015)
17. Belavagi, M.C., Muniyal, B.: Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Comput. Sci. **89**, 117–123 (2016)

18. Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., Dai, K.: An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst. Appl. **39**(1), 424–430 (2012)
19. Kumar, G.: AI based supervised classifiers : an analysis for intrusion detection. In: Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, pp. 170–174 (2011)
20. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA, 1–6 (2009)
21. Dhanabal, L., Shantharajah, S.P.: A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. **4**(6), 446–452 (2015)
22. Zamani, M., Movahedi, M.: Machine learning techniques for intrusion detection. Comput. Sci. **2**, 1–11 (2015)
23. Sharma, R.K., Kalita, H.K., Borah, P.: Analysis of machine learning techniques based intrusion detection systems á supervised learning. In: Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, vol. 44, pp. 485–493 (2016)
24. Chowdhury, M.N., Ferens, K., Ferens, M.: Network intrusion detection using machine learning. Int. Conf. Secur. Manag. **4**, 30–35 (2010)
25. Hamid, Y.: Machine learning techniques for intrusion detection : a comparative analysis. In: ICIA, vol. 7, pp. 0–5. ACM (2016)
26. Ambusaidi, M., He, X., Nanda, P., Tan, Z.: Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans. Comput. **65**(10), 2986–2998 (2016)
27. Singh, R., Kumar, H., Singla, R.K.: An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. **42**(22), 8609–8624 (2015)
28. Proti, D.D.: Review of KDD cup, NSL-KDD and kyoto, datasets. Mil. Tech. Cour. **66**, 580–596 (2006)
29. Angelo, P., Resende, A., Drummond, A.C.: A survey of random forest based methods for intrusion detection systems. ACM Comput. Surv. **51**(3), 52:1–52:27 (2018)
30. Devaraju, S., Ramakrishnan, S.: Performance analysis of intrusion detection system using various neural network classifiers. Int. Conf. Recent Trends Inf. Technol. ICRTIT **2011**, 1033–1038 (2011)