Empirical evaluation of Machine Learning models for Intrusion Detection

Tomislav Takač, Goran Sladić, Aleksandar Kovačević, Jelena Slivka
Faculty of Technical Sciences,
Computing and Control Engineering,
Chair of Informatics, Novi Sad, Serbia
takac.dp32.2019@uns.ac.rs, sladicg@uns.ac.rs, kocha78@uns.ac.rs, slivkaje@uns.ac.rs

Abstract – Intrusion detection is vital in ensuring network security. Recently, researchers started focusing on developing machine learning (ML) techniques for intrusion detection as they promise to solve the problems of other techniques: high false positive alarm rates and the inability to detect unknown attacks. For the ML-based intrusion detection model to be useful in practice, it needs to be trained on the dataset that reflects a realistic network setting. Furthermore, as attacks evolve rapidly, it is crucial to use recent datasets that reflect current network environments. Recently, a publicly available dataset UNSW-NB15 was published. This dataset may serve as a recent realistic benchmark for training and evaluating ML models for intrusion detection. Most existing ML-based intrusion detection solutions have been evaluated on older benchmarks, and it is essential to evaluate how these approaches perform on UNSW-NB15. Thus, in this paper, we examine the literature to find which ML algorithms are frequently used for intrusion detection and evaluate them on UNSW-NB15. We discuss and compare our results to other papers using the same dataset.

I. Introduction

Intrusion detection is a crucial problem in cyber-security [1]. Intrusion is related to attacks whose purpose is to violate network and computer resources' security components or go around them. The annual report of 2016 from the Asia Pacific Computer Emergency Response Team (CERT) showed a significant increment in the number of intrusions and cyber-attacks over the decade [2]. Also, according to a report from the Malaysia CERT published in 2016, 43% of 9986 malicious attacks involve intrusions during system operating hours [3].

There are multiple ways of securing the network. A firewall can protect the system by basic data packet filtering, but it is not adequate for complete network environment protection [4]. Intrusion detection, coupled with a firewall, is a more efficient way to increase network security [5]. Machine Learning (ML) based intrusion detection approaches promise to solve the deficiencies of other approaches: high false alarm rates and the inability to detect unknown attacks [6].

For the ML-based intrusion detection model to be useful in practice, it needs to be trained on the dataset that reflects a realistic network setting. Furthermore, as attacks evolve rapidly, it is crucial to use recent datasets that reflect current network environments [6][7]. Most existing ML-based approaches for intrusion detection have been evaluated on the dated KDD99 dataset [8], which was later shown to have many defects [7]. Recently, a publicly available dataset UNSW-NB15 was published [7]. This dataset may serve as a recent realistic benchmark for training and evaluating ML models for intrusion detection. Compared to other datasets, it contains more features and more training instances, and its training examples are more realistic [9].

In this paper, we aim to evaluate whether the commonly used approaches for ML-based intrusion detection are as effective in the new network environment. More precisely, we aim to derive: (1) what is the best way to evaluate ML-based approaches for intrusion detection and (2) how well do commonly used ML algorithms tackle the intrusion detection problem as measured on the UNSW-NB15 dataset.

To achieve our first goal, we performed a literature survey to answer the following research questions:

RQ1 Which ML algorithms did researchers use to address the network intrusion detection problem?

RQ2 How did the researchers evaluate their approaches?

By examining the existing literature, we have derived the following conclusions:

- The most popular ML-based algorithms for intrusion detection are Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes (NB), Gradient Boosting Classifier (GBC) and Random Forest (RF).
- Model performance should be evaluated in measures of accuracy and prediction time [5].
- Current literature indicates that SVM and RF achieve best performance as measured in accuracy and prediction times [5].

Based on the findings from the literature survey, we trained and evaluated the most popular ML-based algorithms for intrusion detection on the new UNSW-NB15 dataset to achieve our second goal. As the UNSW-NB15 dataset is highly imbalanced, we also apply SMOTE oversampling of the minority class to improve model performance.

The authors of the UNSW-NB15 dataset have predefined the training set (175,341 records) and the testing sets (82,332 records) [10], and we adopt this experimental setting for our evaluation. We perform the 5-fold cross-validation procedure on

the training set to tune algorithm hyper-parameters. Guided by the recommendations from the literature, we measure the performance of the ML-based intrusion detection models in terms of accuracy, but, as the dataset is highly imbalanced, we also measure recall and ROC-AUC. Finally, as recommended in the literature, we measure model training and testing time. Our experiments indicate that the RF classifier has the best performance of all tested algorithms in terms of accuracy (96.56), recall (98.70), and ROC-AUC (99.51). This result agrees with the recent findings [5][11].

The rest of this paper is composed in the following way. In Section II, we present our conclusions derived from the literature survey we performed to answer our research questions. Here, we discuss the target variables and the datasets used to train ML models in the existing literature. Then, we discuss the performance metrics used for evaluating trained ML models. Lastly, we overview which ML algorithms researchers typically use for intrusion detection. Section III presents our experiments and obtained results. Finally, section IV concludes this paper.

II. State-of-the-art Machine Learning approaches for intrusion detection

This section presents our conclusions derived from the literature survey we performed to answer RQ1 and RQ2. Firstly, we discuss the target variable and the datasets used to train ML models in the existing literature (sections A and B). Then, we overview state-of-the-art ML-based intrusion detection approaches and the performance metrics researchers used to evaluate their approaches (section C).

A. Target variable

Dependent on the dataset, the target variable Y can be:

- binary – each record in the dataset is classified as "intrusion" or "not an intrusion" [12-14].
- multi-category – each record is classified as one of the different intrusion types or as "not an intrusion" [15-17].

This problem is not treated as a multi-label problem, as the situation that one instance belongs to multiple categories simultaneously is impossible.

B. Benchmark datasets used for performance comparison of different ML-based intrusion detection models

To ensure the fair comparison of different ML-based approaches for intrusion detection, we need a standardized publicly available dataset. In the following paragraphs, we describe the standard datasets used as benchmarks for the intrusion detection problem.

1) DARPA 1998

MIT's Lincoln laboratory built the DARPA1998 dataset [18]. This dataset is a widely used benchmark in intrusion detection studies. To construct it, the researchers collected internet traffic for over nine weeks. The dataset contains raw packets (binary TCP dump data) labeled as normal (not an intrusion), denial of service (DOS)", Probe, User to Root (U2R), or Remote to Local (R2L).

2) KDD99

ML models cannot be trained directly on the raw packets of data. Thus, Stolfo et al. [19] created the KDD99 [8] dataset as a processed version of the DARPA 1998 dataset, where each raw packet is represented as a feature vector.

KDD99 consists of approximately 4,900,000 instances, where each instance is represented as a vector of 41 features. These features can be classified into two groups:

- basic features, representing the properties contained in the TCP/IP connection.
- content features, from which the attacks can be recognized according to their behavior and the amount of connection to the target host.

KDD99 is the most widely used benchmark dataset in the literature [20]. Unfortunately, as paper [7] later pointed out, the KDD99 dataset includes defects. There are many redundant and repeated records. Thus, researchers need to filter the dataset carefully before they can use it. As a result, the experimental results from different studies are not always comparable. Secondly, the dataset is unbalanced, making the classification models biased toward the majority classes. Finally, is that the KDD dataset is relatively old and unable to represent the current network environment.

3) NSL-KDD

The NSL-KDD [21] was proposed to overcome the shortcomings of the KDD99 dataset. The records in the NSL-KDD were carefully filtered to remove redundant dataset records. Records of different classes are balanced in the NSL-KDD to address the classification bias problem.

NSL-KDD benchmark enables fair comparison of different ML-based approaches. However, due to the performed processing steps reducing the number of KDD99 records, NSL-KDD contains only a moderate number of records and very few samples of the minority classes. Also, the problem of outdated records which do not accurately represent the current network environment remains.

4) UNSW-NB15

Recently, the University of New South Wales compiled the UNSW-NB15 [22] dataset. To collect it, researchers configured three virtual servers to capture network traffic. UNSW-NB15 dataset includes more types of attacks than the KDD99 dataset, as well as more features. The records of the UNSW-NB15 dataset are labeled as belonging to one of the nine attack categories (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms) or as "normal" (not an attack). A 49-dimensional feature vector describes each record.

The UNSW-NB15 represents a new intrusion detection system dataset. It has been used in some recent studies [7] [23-25], however, its influence is still inferior to that of KDD99.

However, it is necessary to construct new datasets for training ML-based intrusion detection models. Several recent studies [5] [9] [20] showed that the KDD99 dataset and its variants do not reflect the current state of the network threat environment in terms of network traffic modern attacks.

C. State-of-the-art ML-based intrusion detection approaches

We examined the literature to RQ1 to derive which ML algorithms did researchers use to address the network intrusion detection problem. We considered the recent papers applying traditional ML techniques for intrusion detection. In this initial study, we limit our research to the traditional ML models and leave deep learning approaches for future work. While deep learning approaches are state-of-the-art in many domains, they typically require more data and may prove unnecessary if traditional models yield good performance.

To detect intrusions, in [5], the authors apply the following ML algorithms: Support Vector Machine, Naïve Bayes, Decision Tree, and Random Forest. In their experiments, they use the UNSW-NB15 dataset. They evaluate the effectiveness of the considered models in terms of accuracy, sensitivity, specificity, training, and prediction time. In their experiments, the Random Forest classifier achieved the best accuracy (97.49%).

Authors of [26] experiment with Random Forest and Support Vector Machine classifiers. They apply these classifiers to the KDD 99 dataset. Before training, the dataset features are preprocessed: categorical features are label-encoded, and all features are normalized. The authors evaluated the performance of their approach in terms of detection rate and false alarm rate. They concluded that the Random Forest classifier yields the best performance. According to the variable performance assessed by the Random Forest classifier, they selected the top 14 features and achieved a higher Detection rate compared to the considered alternatives.

In the paper [27], the authors filtered the KDD99Train+ and KDD99Test+ datasets to remove the redundant records that existed in the original data. They also apply SVM and RF and the same data processing techniques as [26]. They measure algorithm performance in terms of accuracy, false-negative rate, and precision. In contrast to [26], SVM achieved slightly higher accuracy (92.99%) than RF (91.41%). However, RF requires significantly less training time.

Authors of the paper [28] apply Logistic Regression, Support Vector Machines, Random Forest, Gradient Boosted Decision trees, and Naive Bayes on the KDD 99 and the NSL-KDD datasets. They experiment with two feature selection techniques: correlation-based and Chi-Squared feature selection. They evaluate the performance of their approach in terms of accuracy, sensitivity, specificity, and training and prediction time. They conclude that removing highly correlated features from the NSL-KDD dataset increases the accuracy of Random Forest and Gradient Boosted Decision trees. However, feature selection decreased the accuracy of Logistic Regression, SVM, and Naïve Bayes by a small margin.

Paper [11] considers only the Random Forest classifier. The authors experimented with different values for the number of trees used in the RF classifier and run their experiments on three datasets: NSL-KDD, UNSW-NB15, and GPRS. They measured algorithm performance in terms of accuracy and false-positive rate. They concluded that Random Forest with 800 trees results is statistically significant accuracy and false-positive rate increase compared to other classifiers. The accuracy achieved by the RF-800 classifier on the UNSW-NB15 dataset was 95.5%.

To summarize, in the existing literature, the most frequently used ML algorithms for tackling the intrusion detection problem are Support Vector Machine (SVM), Gradient Boosting Classifier (GBC), Random Forest, Naïve Bayes, and K-Nearest Neighbors (KNN) [5] [11] [26-28]. SVM and Random Forest have proved to have high performance [5] [27]. Most of the published papers and works use the KDD99 dataset and its variants and report high performances. However, KDD99 has reported shortcomings. The UNSW-NB15 dataset more realistically simulates actual network intrusion compared to these earlier benchmarks. However, very few papers evaluate their approaches on this dataset.

III. VERIFICATION

Based on the examined literature (section II), we derived that the most used performance measure for ML-based intrusion detection models is accuracy. Accuracy is computed as the ratio of the correct classification examples of all the classification examples:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN},$$

where:

- TP stands for True Positives (instances of attacks correctly detected as attacks),
- TN stands for True Negative (typical network behavior correctly recognized as typical network behavior),
- FP stands for False Positive (typical network behavior falsely recognized as an attack), and
- FN stands for False Negative (attack falsely recognized as typical network behavior).

As the UNSW-NB15 dataset is imbalanced, we also report the achieved recall and ROC-AUC as the measure of performance of ML-based intrusion detection algorithms. Recall (R) is computed as the ratio of the true positive predictions in the total amount of positive predictions:

$$R = \frac{TP}{TP + FN}.$$

To define ROC-AUC, we first define True positive rate (TPR) as the ratio of true positive predictions to all positive predictions:

$$TPR = \frac{TP}{TP + FN},$$

and False positive rate (FPR) as the ratio of false positive predictions to all negative predictions:

$$FPR = \frac{FP}{FP + TN}.$$

In ROC-AUC, ROC denotes a probability curve, and AUC (Area Under Curve) denotes the degree or measure of separability. The ROC curve is plotted with TPR against the FPR, where TPR is on the y-axis and FPR is on the x-axis. ROC-AUC is a measure of how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is.

Finally, some authors emphasize that the model prediction time [5] should also be considered when comparing different ML-based intrusion detection models. Thus, we perform all experiments on Windows 10 Pro, Intel(R) Core(TM) i7-6700HQ 2.6GHz, 16 GB RAM, and report the prediction time for each model.

The authors of the UNSW-NB15 dataset have predefined the training set (175,341 records) and the testing sets (82,332 records) [10], and we adopt this experimental setting for our evaluation. Using the available training data, we train the following ML models derived as the most popular and successful ML-based intrusion detection models used in the literature:

- Support Vector Machine (SVM)
- K-Nearest Neighbor (KNN)
- Naïve Bayes (NB)
- Gradient Boosting Classifier (GBC)
- Random Forest (RF)

Since the USNW-NB15 dataset is highly imbalanced, we also apply the SMOTE [29] oversampling of the minority class. We made sure to perform oversampling on the training set and leave the test set intact.

We perform the 5-fold cross-validation procedure on the training set to tune the hyper-parameters of each algorithm. To tune the hyper-parameters, we first perform a randomized search to obtain a good parameter combination. We then further refined the parameters by applying the grid search.

Table 1 presents the obtained results. In our experiments, the RF classifier performs better than its alternatives in terms of Accuracy, Recall, ROC-AUC. The second-best classifier in terms of these performance measures is the GBC. Compared to other studies, the RF and NB classifiers achieved similar accuracy as reported in [5][11]. However, in our study, SVM achieved lower accuracy than reported in [5].

The NB classifier was the fastest algorithm in terms of prediction time, followed by KNN and RF. SVM has the longest prediction time. These results agree with [5].

The very high performance achieved by the Random Forest model (ROC-AUC of 99.51), together with its high prediction speed (7 s), indicates that the ML approach for intrusion detection can be successfully applied in a realistic setting.

TABLE 1. COMPARING PERFORMANCE OF MULTIPLE ALGORITHMS ON UNSW-NB15 DATASET

Algorithm	Accuracy	Recall	ROC-AUC	Training time [s]	Prediction time [s]
SVM	78.64	93.22	92.43	4765.76	41.8
KNN	88.31	93.54	93.91	11.10	8.78
NB	78.24	85.39	90.13	**4.05**	**1.54**
GBC	92.82	95.82	94.81	456.28	2.92
RF	**96.56**	**99.31**	**99.51**	262.31	7.05

IV. CONCLUSION

This paper tackled the intrusion detection problem by experimenting with several ML-based algorithms on the UNSW-NB15 dataset that reflects a realistic modern network setting. We focused on the binary classification task where each dataset record is classified either as an intrusion or as regular network traffic. The ML-based intrusion detection models' high performance and prediction speed achieved in our experiments indicate that these algorithms can be successfully applied for intrusion detection in a realistic current network environment.

In the future, we aim to experiment with different feature selection and dimensionality reduction techniques to reduce training and detection time. We also plan to experiment with more fine-grained results in the form of multi-class classification, where the “intrusion” category can be further divided into nine categories.

REFERENCES

[1] Asif, M.K., Khan, T.A., Taj, T.A., Naeem, U. and Yakoob, S., 2013, April. Network intrusion detection and its strategic importance. In 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC) (pp. 140-144). IEEE. Asif, M.K., Khan, T.A., Taj, T.A., Naeem, U. and Yakoob, S., 2013, April. Network intrusion detection and its strategic importance. In 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC) (pp. 140-144). IEEE.

[2] Asia Pacific Computer Emergency Response Team. Available online: http://www.apcert.org/ (accessed on20 September 2016)

[3] Malaysia Computer Emergency Response Team Incident Statistics. Available online: http://www.mycert.org.my/en/ (accessed on 20 September 2016).

[4] A Short Survey of Intrusion Detection Systems Vera Marinova-Boncheva, PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 2007

[5] Mustapha Belouch, Salah El Hadaj, Mohamed Idhammad, Performance evaluation of intrusion detection based on machine learning using Apache Spark

[6] Liu, H. and Lang, B., 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), p.4396.

[7] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems

Conference (MilCIS), Canberra, ACT, 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.
[8] KDD99 Dataset. 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 29 Jun 2020).
[9] Tharmini Janarthanan, Shahrzad Zargari, Feature Selection in UNSW-NB15 and KDDCUP'99 datasets, 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)
[10] UNSW-NB15 Dataset https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (29.07.2020)
[11] Primartha, Rifkie, and Bayu Adhi Tama. "Anomaly detection using random forest: A performance revisited." 2017 International conference on data and software engineering (ICoDSE). IEEE, 2017.
[12] Vandenberghe, Rik, et al. "Binary classification of 18F-flutemetamol PET using machine learning: comparison with visual reads and structural MRI." NeuroImage 64 (2013): 517-525.
[13] Kirichenko, Lyudmyla, Tamara Radivilova, and Vitalii Bulakh. "Binary Classification of Fractal Time Series by Machine Learning Methods." International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence". Springer, Cham, 2019.
[14] Nawir, Mukrimah, et al. "Performances of machine learning algorithms for binary classification of network anomaly detection system." Journal of Physics: Conference Series. Vol. 1018. 2018.
[15] Polat, Kemal, and Salih Güneş. "A novel hybrid intelligent method based on C4. 5 decision tree classifier and one-against-all approach for multi-class classification problems." Expert Systems with Applications 36.2 (2009): 1587-1592.
[16] Panca, V., and Zuherman Rustam. "Application of machine learning on brain cancer multiclass classification." AIP Conference Proceedings. Vol. 1862. No. 1. AIP Publishing LLC, 2017.
[17] Rahman, Md Mahmudur, Bipin C. Desai, and Prabir Bhattacharya. "Medical image retrieval with probabilistic multi-class support vector machine classifiers and adaptive similarity fusion." Computerized Medical Imaging and Graphics 32.2 (2008): 95-108.
[18] DARPA1998 Dataset. 1998. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 29 Jun 2020).
[19] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000
[20] Divekar, Abhishek, et al. "Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives." 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). IEEE, 2018.
[21] NSL-KDD99 Dataset. 2009. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 29 Jun 2020).
[22] UNSW-NB15 Dataset. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed on 29 Jun 2020).
[23] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective 25.1-3 (2016): 18-31.
[24] N. Moustafa, J. Slay and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," in IEEE Transactions on Big Data, vol. 5, no. 4, pp. 481-494, 1 Dec. 2019, doi: 10.1109/TBDATA.2017.2715166.
[25] Moustafa, Nour, Gideon Creech, and Jill Slay. "Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models." Data analytics and decision support for cybersecurity. Springer, Cham, 2017. 127-156.
[26] Chang, Yaping, Wei Li, and Zhongming Yang. "Network intrusion detection based on random forest and support vector machine." 2017 IEEE intl. conf. on computational science and engineering (CSE) and IEEE intl. conf. on embedded and ubiquitous computing (EUC). Vol. 1. IEEE, 2017.
[27] Hasan, Md Al Mehedi, et al. "Support vector machine and random forest modeling for intrusion detection system (IDS)." Journal of Intelligent Learning Systems and Applications 2014 (2014).
[28] Gupta, Govind P., and Manish Kulariya. "A framework for fast and efficient cyber security network intrusion detection using apache spark." Procedia Computer Science 93 (2016): 824-831.
[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer,"SMOTE: Synthetic Minority Over-Sampling Technique,"Journal ofArtificial Intelligence Research, vol. 16, pp. 321–357, 2002