Anomaly Detection with Various Machine Learning Classification Techniques over UNSW-NB15 Dataset

Martina Shushlevska, Danijela Efnusheva, Goran Jakimovski and Zdravko Todorov
Computer Science and Engineering Department, Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, 18 Rugjer Boshkovik Str., Skopje, R. N. Macedonia
martinasuslevska@yahoo.com, {danijela, goranj, todorovz}@feit.ukim.edu.mk

Keywords: Anomaly Detection, Intrusion Detection System, Machine Learning, Network Security, UNSW-NB15 Dataset.

Abstract: The exponential growth of computers and devices connected to the Internet and the variety of commercial services offered creates the need to protect Internet users. As a result, intrusion detection systems (IDS) are becoming an essential part of each computer-communication system, detecting and responding to malicious network traffic and computer abuse. In this paper, an IDS based on the UNSW-NB15 dataset has been implemented. The results obtained indicate F1 Score and Recall values of 76.1% and 85.3% for the Naive Bayes algorithm, 78.2% and 96.1% for Logistic Regression algorithm, 88.3% and 95.4% for Decision Tree classifier, and 89.3% and 98.5% for Random Forest.

1 INTRODUCTION

Network attacks are one of the biggest security problems in the world today. The constant increase in computers, mobile phones, sensors, IoT devices, big data, web applications, server and cloud systems, and more sophisticated computing resources imposes even more significant challenges for keeping network connections under control. Additionally, the enormous increase in network traffic has caused many new approaches to network intrusions to be planned by various hackers and malicious users. Therefore, IDS are a rapidly evolving field aimed at providing detection of malicious behaviour and attacks in the network [1].

The two crucial methods for detecting threats that intrusion detection systems can use are: signature-based and anomaly-based [1]. Signature-based detection is usually applied in identifying known threats, by using a pre-programmed list of them and their indicators of compromise. In fact, an indicator of compromise could be a specific behaviour that generally precedes a malicious network attack, known byte sequences, malicious domains, file hashes, or even the content of email subject headings. On the other hand, an anomaly-based IDS is used to alert a suspicious behaviour that is unknown. An anomaly-based detection system doesn't operate by searching for known threats, but it may utilize machine learning methods for training the detection system to recognize a normalized baseline. This baseline shows what is the system's normal behaviour, and then all network activity is compared to that baseline. Therefore, instead of searching for known indicators of compromise, an anomaly-based IDS identifies any odd behaviour in order to trigger alerts.

Many techniques have been developed to detect anomaly-based intrusions by applying data mining and machine learning methods [2-7]. Mainly, well-known datasets (ex. KDDCUP'99, NLS-KDD, UNSW-NB15) that consist of real-time network traffic with a large number of features are used in anomaly-based intrusion detection [8], [9]. Therefore, in this paper, we implement several ML algorithms over the UNSW-NB15 dataset to analyse and verify that machine learning is very applicable for solving a problem with unauthorized attacks in network traffic. Assuming that KDDCUP'99 and NSL-KDD benchmark datasets were generated a decade ago, in this research, we use the UNSW-NB15 dataset that was published in 2015. This dataset targets more realistic and network traffic and novel types of modern attacks. Indeed, UNSW-NB15 is a network intrusion dataset that contains raw network packets, characterized with 49 features and organized in 10 categories (9 attack types plus 1 for normal activity) [9].

This paper aims to examine the differences between a Naive Bayes (NB), a Logistic Regression (LR), a Decision Tree (DT), and a Random Forest (RF) ML algorithms in order to determine the strengths and weaknesses of using these methods over the UNSW-NB15 dataset. By evaluating the performance of these algorithms in terms of accuracy, precision, recall, and F1 metrics, we can consider which of the analysed classification methods is the most effective and suitable for detecting anomalies.

The rest of the paper is organized as follows: Section II presents the state of the art in the domain of intrusion detection. Section III provides a brief description of the UNSW-NB15 dataset and explains how the ML model is built. Section IV analyses the results from several classification methods, including Naive Bayes, Logistic Regression, Decision Tree, and Random Forest. Section V concludes the paper and provides directions for future work.

2 CURRENT STATE

As more people use the Internet for personal or business reasons, different cyber-attacks and intrusions grow daily. An IDS is one of the most crucial considerations of cyber-security. This type of system can be software or hardware-based and can recognize successful violations even after they have happened. Generally, an IDS’s purpose is to monitor network packets or systems to detect malicious activity and take specific measures [1].

There are many types of IDSs, which are discussed and summarized below.

A host-based IDS (HIDS) monitors and analyzes the internal computing system or system-level activities of a single host such as system configuration, application activity, wireless network traffic (only for that host) or network interface, system logs or audit log, running user or application processes, file access and modification security logs [10]. Examples of some known HIDS systems are Tripwire and OSSEC.

A network-based IDS (NIDS) monitors and analyzes network traffic on specific network segments for suspicious activities detection. This type of IDS is activated when packets enter a particular network from the Internet, and its function is to decide whether to reject or accept the entry packets and pass them to the local network. An example of a known NIDS system is Snort [11].

A protocol-based IDS (PIDS) monitors and checks the specific protocol behavior and its state like HyperText Transfer Protocol (HTTP). It focuses on actions in some particular application by monitoring and analyzing the application log files or measuring their performance. A PIDS approach for detecting jamming attacks in a LoRaWAN network is proposed in [12].

A wireless IDS (WIDS) monitors wireless networks to detect any harmful activity (ex. too many de-authentication packets, too many broadcast requests, analysis of the number of packets sent during a single time window). If malicious behavior from certain users is detected, they forbid them from connecting to the wireless network access point. Examples of some known WIDS systems are Kismet and NetStumbler [13].

Network behavior analysis (NBA) monitors and checks network traffic to detect threats that produce uncommon traffic flows, such as DDOS attacks, malware, and policy violations [14]. It is recommended to be used together with a firewall and other types of IDS systems.

Nowadays, due to increased use of the Internet and company networks, network traffic increases daily. Access to company networks should be given only to authorized users, so, detecting unauthorized entities or intruders is necessary. Machine learning techniques have been used and applied in many studies [2-7], where they have provided solid results in detecting intrusions and protecting the network from sudden attackers. The applicability of ML for intrusion detection systems is due to well-known technologies, such as identification, extraction, classification, regression, and prediction, as well as solid datasets composed of real-time network traffic with many features and their description. For example, the research in [2] and [3] gives an opportunity to review classification techniques and ML models for an IDS application.

There is also a hybrid attack detection system based on SVM (Support Vector Machine) and C5.0 Decision Tree proposed by authors in [4], where using a combination of popular ML algorithms improves the accuracy of attack detection, compared to being used apart. A similar hybrid system in which two algorithms (K-means and NB) are used to group some data and classify it is proposed by authors in [5]. MapReduce is very popular for processing extensive structured and unstructured data placed in key/value pairs. The authors of [6] propose an intrusion detection model that uses

MapReduce. MapReduce relies on using a combination of Fuzzy C-means (FCM) and SVM for classification and generating key pairs/values for attack detection. Furthermore, a survey of different approaches for intrusion detection with deep learning is given in [7].

3 OVERVIEW OF UNSW_NB15 DATASET AND BUILDING ML MODELS

UNSW-NB15 is a network traffic dataset with different categories for normal activities and malicious attacks, generated by the Australian Center for Cyber Security and published in 2015, [9]. This dataset includes 100 GB of raw network traffic (pcap files) generated as a hybrid of real normal activities and synthetic contemporary attack behaviors. Indeed, the traffic is categorized into nine different attacks and a wide range of real normal activities. The complete dataset contains 257,673 records, each represented by 49 features and a class label.

The following text discusses the nine types of attacks that are included in the UNSW-NB15 dataset:

1) Analysis: a type of attack where the attacker listens to the network traffic and then performs analysis of the observed data.
2) Backdoors: a type of attack that provides attackers with unauthorized remote access to a system without the usual authentication process.
3) DoS: a type of attack in which the attacker crashes or floods the services of a target machine, in order to make it overloaded and unavailable for serving further requests.
4) Exploit: a type of attack which utilizes the software vulnerabilities and errors within the networks, operating systems or hardware.
5) Fuzzers: a type of attack in which the attacker tries to stress the application in order to cause unexpected behavior, such as resource leaking or even crashes.
6) Generic: a type of attack that acts against a cryptographical primitive and it tries to break the key of some secure system.
7) Reconnaissance: a type of attack that gathers information about the target computer network in order to bypass its security control. Some examples are: phishing, social engineering port scanning, packet sniffing, etc.
8) Shellcode: a type of malware attack in which the attacker uses a special type of code that is used to exploit a variety of software vulnerabilities, so the attacker could take control over the compromised machine.
9) Worms: a type of malware attack that replicates itself in order to be spreaded to other computers by a computer network.

The most common attacks in the UNSW-NB15 database are Generic and Exploits, with are a total of 40000 and 33393 records, respectively. Additionally, if an analysis of the number of malicious or normal dataset records is made, we get the distribution shown in Figure 1. Here it can be seen that there is a higher prevalence of malicious records (68.06%) compared to the prevalence of normal traffic records (31.94%). Malicious records include the nine types of previously described attacks.

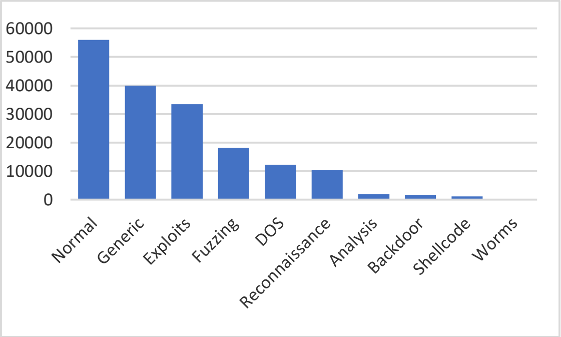

Figure 1: Number of records that represent normal traffic and malicious types of attacks in the UNSW-NB15 dataset.

The UNSW-NB15 dataset is characterized with 49 features shown in Table 1. These features are organized in six groups, discussed below:

1) Flow features (0-5): These features have the identifier attributes between hosts (client-server and vice-versa).
2) Basic features (6-18): These features include the attributes that represent protocols connections.
3) Content features (19-26): These features contain the attributes of TCP/IP and as well some attributes of http services.
4) Time features (27-35): This group contains the attributes of time, such as: start/end packet time, arrival time between packets, and round trip time of TCP protocol.
5) Additional generated features (36-47). This group can be further divided into two groups:

- General purpose features (36-40), whereas each feature of this group has its own purpose, in order to protect the service of protocols.
- Connection features (41-47) are built from the flow of 100 record connections based on the sequential order of the last time feature.

6) Labelled Features (48-49): This category shows the label and attack type of each record.

Table 1: UNSW-NB15 dataset features.

N	Feature	Description
1	srcip	Source IP address
2	sport	Source port number
3	dstip	Destination IP address
4	dsport	Destination port number
5	proto	Protocol type
6	state	The state
7	dur	Row total duration
8	sbytes	Source to destination bytes
9	dbytes	Destination to source bytes
10	sttl	Source to destination time to live
11	dttl	Destination to source time to live
12	sloss	Source packets retransmitted or dropped
13	dloss	Dest. packets retransmitted or dropped
14	service	Such as http, ftp etc.
15	sload	Source bits per second
16	dload	Destination bits per second
17	spkts	Source to dest. packet count
18	dpkts	Dest. to source packet count
19	swin	Source TCP window adv. value
20	dwin	Source TCP window adv. value
21	stcpb	Source TCP base seq. num.
22	dtcpb	Dest. TCP base seq. num.
23	smeansz	Mean of the packet size transmitted by the srcip
24	dmeansz	Mean of the packet size transmitted by the dstip
25	trans_depth	The connection of http req./resp. transaction
26	res_bdy_len	The content size of the data transferred from http
27	sjit	Source jitter
28	djit	Destination jitter
29	stime	Row start time
30	ltime	Row last time
31	sintpkt	Source inter-packet arrival time
32	dintpkt	Dest. inter-packet arrival time
33	tcprtt	Setup round trip time
34	synack	Time between SYN and SYN_ACK packets
35	ackdat	Time between SYN_ACK and ACK packets
36	is_srn_ips_ports	If srcip(1)=dstip(3) and sport(2)=dsport(4), assign 1else 0
37	ct_state_ttl	No. for each state (6) according to values of sttl(10) and dttl(11)
38	ct_flw_http_mthd	No. of fows that has methods like Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of fows that has a command in ftp session.
41	ct_srv_src	No. of rows of the same service(14) and srcip(1) in 100 rows
42	ct_srv_dst	No. of rows of the same service(14) and dstip(3) in 100 rows
43	ct_dst_ltm	No. of rows of the same dstip(3) in 100 rows
44	ct_src_ltm	No. of rows of the same srcip(1) in 100 rows
45	ct_src_dport_ltm	No. of rows of the same srcip(1) and the dsport(4) in 100 rows
46	ct_src_sport_ltm	No. of rows of the same dstip(3) and the sport(2) in 100 rows
47	ct_dst_src_ltm	No. of rows of the same srcip(1) and the dstip(3) in 100 rows
48	attack_cat	Type of attack
49	label	0 for normal and 1 for attack

Python is used to process the UNSW-NB15 dataset in conjunction with the Jupyter Notebook tool that is an open-source web application used for generating and sharing documents which contain live code, equations, visualizations, and text. In particular, the following Python libraries [15] have been used in the analysis, processing and creation of the classification models: Pandas, NumPy, matplotlib.pyplot, Seaborn and sklearn (Scikit-learn).

The UNSW-NB15 dataset is defined by two files, a training set and a testing set (UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv respectively). The training set includes 175,341 records, while the testing set includes 82,332 records. Accordingly, 31.95% of the records belong to the testing set, and 68.05% of the records belong to the training set. Each record can represent some of the nine types of attacks or normal traffic.

5-Fold Cross-Validation is used to build an ML model, so the UNSW-NB15 dataset is divided into 5 parts. In the first iteration, the first section is used to validate the model, and the rest (the other 4 sections) are used to train the model. In the second iteration, the second division is used as a validation set, while the others serve as training sets. This process is repeated until each of the five divisions is used as a validation set. This method is used for building an ML model for each of the analyzed algorithms, including: Naive Bayes, Logistic Regression, Decision Tree and Random Forest.

4 ANALYSIS OF RESULTS

In this paper, a research is done that develops a system for detecting attacks by differentiating anomalies from normal data flow based on network behavior. One advantage of this approach is that when an attack occurs, the network behavior will deviate from the normal pattern of behavior and the anomaly will be detected. In order to avoid the effect of data sampling when assessing the IDS, 5-fold cross-validation (CV) method is used. Four different machine learning algorithms (NB, LR, DT and RF) are applied on the dataset.

The analyzed metrics of the experiment are: CV fit time, CV accuracy mean, CV precision mean. CV recall mean, CV F1 mean, CV AUC mean, Accuracy test, Precision test, Recall test, F1 test and AUC test. The CV fit time refers to the required time for fitting the estimator on the train set for each of the five CV splits. In fact, the performance CV metrics reported by 5-fold cross-validation are calculated as an average of the values computed in 5 steps. In each of the steps, the model is trained using 4 of the folds as training data, and validated with the remaining part of the data. After that, final evaluation is done on the testing set, by measuring the Accuracy test, Precision test, Recall test, F1 test and AUC test values. Accuracy identifies how many observations, both positive and negative, were properly classified. Precision represents the ratio of properly predicted positive observations to the total predicted positive observations. Recall is the ratio of properly predicted positive observations to the all observations in an actual class. F1 Score combines precision and recall in one metric by calculating the harmonic mean between them. AUC is the area under the ROC (Receiver Operator Characteristic) curve, which is used to show the diagnostic ability of binary classifiers. The results can be seen on Table 2.

Table 2: Analysis of anomaly detection with NB, LR, DT, and RF classification algorithms over UNSW-NB15 dataset.

Metric	NB	LR	DT	RF
CV fit time [s]	0.37489	2.12689	2.85655	59.11744
CV accuracy mean	0.79568	0.85093	0.94884	0.95991
CV precision mean	0.84303	0.83782	0.96293	0.96320
CV recall mean	0.85993	0.96845	0.96186	0.97848
CV F1 mean	0.85136	0.89841	0.96239	0.97078
CV AUC mean	0.86780	0.86922	0.94299	0.99354
Accuracy test	0.70620	0.70598	0.86123	0.87093
Precision test	0.68783	0.65981	0.82205	0.81771
Recall test	0.85396	0.96199	0.95460	0.98522
F1test	0.76195	0.78275	0.88338	0.89368
AUC test	0.79999	0.81454	0.85322	0.97730

The results given in Table 1 show that the Random Forest algorithm gives better results for each of the analyzed metrics, during the model validation and testing, compared to the results obtained for the other three algorithms (exception is Test Precision).

The following text provides a more thorough evaluation of the results for the CV Fit Time, F1 test and Recall test parameters obtained for the observed algorithms (NB, LR, DT and RF).

4.1 Analysis of CV Fit Time Metric

Figure 2 shows the CV Fit Time for each of the four algorithms (NB, LR, DT and RF). According to that, it can be seen that the learning time of Random Forest is 20-30 times longer than for the other algorithms (LR and DT). Also it can be seen that Naive Bayes classifier has very small CV Fit Time, that is 158 times faster than the one attained for the Random Forest algorithm.

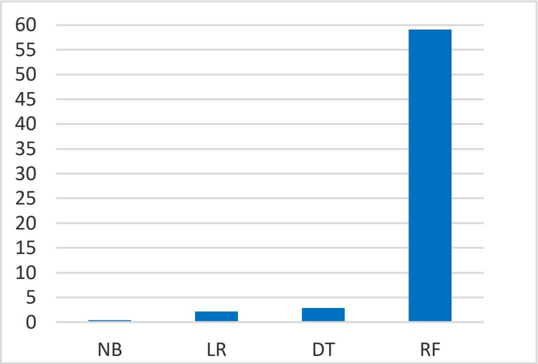

Figure 2: Comparison of CV Fit Time [sec] for NB, LR, DT and RF algorithms, implemented over UNSW-NB15 dataset.

4.2 Analysis of F1 Test Metric

The created anomaly detection model should have a relatively high coverage capability and high accuracy. Accordingly, the F1 Test is selected as the assessment metric. The F1 result can be interpreted as the average of precision and recall, where the F1 result reaches its best value when it is 1 and its worst result when it is 0. Figure 3 shows the F1 Test results for each of the four algorithms. From there it can be seen that all the algorithms (NB, LR, DT and RF) are more close to the 1, so all of them are valid and acceptable models. However, for the analyzed UNSW-NB15 dataset the Random Forrest algorithm is the best basic model for classification.

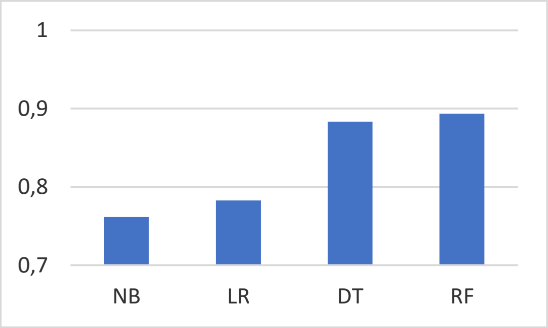

Figure 3: Comparison of F1 Test metric for NB, LR, DT and RF algorithms, implemented over UNSW-NB15 dataset.

4.3 Analysis of Recall Test Metric

When an unbalanced classification problem for anomaly detection is being analyzed, the recall metric should be observed as well. This metric is used to determine how many of the classified attacks were a real attack. Figure 4 shows the values of Recall Test metrics for each of the four algorithms. Accordingly, it can be noticed that Random Forest provides the best Recall Test result of 0.985 (i.e. 98,5%), but also Logistic Regression algorithm is very close to achieve the maximal score, given that its Recall Test result is 0,961 (i.e. 96,1%).

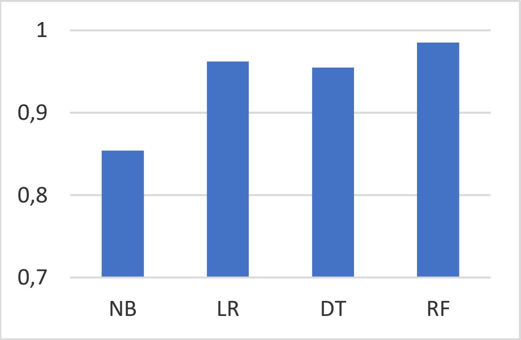

Figure 4: Comparison of Recall Test metric for NB, LR, DT and RF algorithms, implemented over UNSW-NB15 dataset.

5 CONCLUSIONS

This paper presents an implementation of IDS based on the UNSW-NB15 dataset. The dataset is trained and tested for nine class attack categories. With the application of four machine learning algorithms, Naive Bayes, Logistic Regression, Decision Tree, and Random Forrest, the UNSW-NB15 dataset has been successfully classified into network traffic of normal records and attack logs. From the analysis of the ML models for each of the methods, it was shown that the classification with Random Forrest is more successful than with Naive Bayes, Logistic Regression, and Decision Tree. According to the obtained results, the Random Forest classifier provides F1 and Recall values of 89.3% and 98.5%. The good results of Random Forrest training indicate that this algorithm requires far less need to find hyper-parameters, which are left as default. On the other hand, the Naive Bayes classifier shows the least effectiveness when applied in the UNSW-NB15 data set. In order to provide more extensive analysis, other ML classification algorithms and feature selectors could be applied to the UNSW-NB15 data set in the future.

REFERENCES

[1] L. H. Yeo, X. Che, and S. Lakkaraju, “Understanding modern intrusion detection systems: a survey,” in Cryptography and Security Journal, 2017.

[2] P. Amudha, S. Karthik, and S. Sivakumari, “Classification techniques for intrusion detection-an overview,” in International Journal of Computer Applications, vol. 76, no. 16, 2013.

[3] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, “Evaluation of machine learning algorithms for intrusion detection system,” in Proc. of IEEE International Symposium on Intelligent Systems and Informatics, 2017.

[4] V. Golman, “An efficient hybrid intrusion detection system based on C5.0 and SVM,” in International Journal of Database Theory and Application, vol. 7, no. 2, 2014, pp. 59-70.

[5] S. S. Tanpure, G. D. Patel, Z. Raja, J. Jagtap, and A. Pathan, “Intrusion detection system in data mining using hybrid approach,” in International Journal of Computer Applications, 2016, pp. 0975-8887.

[6] S. A. Hajare, “Detection of network attacks using big data analysis,” in International Journal on Recent and Innovation Trends in Computing and Communication, vol. 4 (5), 2016, pp. 86-88.

[7] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study,” in Journal of Information Security and Applications, vol. 50, 2020.

[8] D. D. Protić, “Review of KDD CUP ‘99, NSL-KDD and KYOTO 2006+ datasets,” in Military Technical Courier, vol. 66 (3), 2018.

[9] M. Nour, J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. of IEEE Military Communications and Information Systems Conference, 2015.

[10] S. M. Othman, N. T. Alsohybe, F. M. Ba-Alwi, and A. T. Zahary, “Survey on intrusion detection system types,” in International Journal of Cyber-Security and Digital Forensics, vol. 7, no. 4, 2018, pp. 444-462.

[11] B. Caswell, J. Beale, and A. Baker: Snort Intrusion Detection and Prevention Toolkit. MA, Burlingthon: Syngress, 2007.

[12] S. Danish, A. Nasir, H. K. Qureshi, A. B. Ashfaq, S. Mumtaz, and J. Rodriguez, “Network intrusion detection system for jamming attack in LoRaWAN join procedure,” in Proc. of IEEE International Conference on Communications, 2018.

[13] K. Hutchison, “Wireless intrusion detection systems,” SANS Institute, White Paper, 2005.

[14] W. Stallings, “Network security essentials: applications and standards”, 6th ed. USA: Pearson, 2017.

[15] S. Madhavan, “Mastering python for data science”, UK: Packt Publishing, 2015.

[16] R. Ioshi, “Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures,” 2016, [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.