2021-11-17

Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach

Agghey, Abel

MDPI

https://doi.org/10.3390/sym13112192

symmetry

MDPI

Article

Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach

Abel Z. Agghey [1,*], Lunodzo J. Mwinuka [2], Sanket M. Pandhare [3], Mussa A. Dida [1] and Jema D. Ndibwile [4,*]

[1] School of Computation and Communication Science and Engineering, The Nelson Mandela African Institution of Science and Technology, Arusha 23311, Tanzania; mussa.ally@nm-aist.ac.tz
[2] Computing Science Studies, Mzumbe University, Morogoro 67311, Tanzania; lumwinuka@mzumbe.ac.tz
[3] Center for Excellence in Information Technologies, Pune 411008, India; sanketp@cdac.in
[4] College of Engineering, Carnegie Mellon University Africa, Kigali BP 6150, Rwanda
* Correspondence: aggheya@nm-aist.ac.tz (A.Z.A.); jndibwil@andrew.cmu.edu (J.D.N.)

Abstract: Over the last two decades (2000–2020), the Internet has rapidly evolved, resulting in symmetrical and asymmetrical Internet consumption patterns and billions of users worldwide. With the immense rise of the Internet, attacks and malicious behaviors pose a huge threat to our computing environment. Brute-force attack is among the most prominent and commonly used attacks, achieved out using password-attack tools, a wordlist dictionary, and a usernames list—obtained through a so-called an enumeration attack. In this paper, we investigate username enumeration attack detection on SSH protocol by using machine-learning classifiers. We apply four asymmetrical classifiers on our generated dataset collected from a closed-environment network to build machine-learning-based models for attack detection. The use of several machine-learners offers a wider investigation spectrum of the classifiers' ability in attack detection. Additionally, we investigate how beneficial it is to include or exclude network ports information as features-set in the process of learning. We evaluated and compared the performances of machine-learning models for both cases. The models used are k-nearest neighbor (K-NN), naïve Bayes (NB), random forest (RF) and decision tree (DT) with and without ports information. Our results show that machine-learning approaches to detect SSH username enumeration attacks were quite successful, with KNN having an accuracy of 99.93%, NB 95.70%, RF 99.92%, and DT 99.88%. Furthermore, the results improve when using ports information.

Keywords: SSH; username enumeration; enumeration attack; password enumeration; brute-force attack; machine-learning

Citation: Agghey, A.Z.; Mwinuka, L.J.; Pandhare, S.M.; Dida, M.A.; Ndibwile, J.D. Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach. *Symmetry* **2021**, *13*, 2192. https://doi.org/10.3390/sym13112192

Academic Editor: Alexander Shelupanov

Received: 21 October 2021
Accepted: 4 November 2021
Published: 17 November 2021

1. Introduction

The Internet is widely recognized for its rapid growth and tremendously usage in current years [1]. As a result, there are symmetrical and asymmetrical Internet consumption patterns. Over four billion individuals have Internet access and utilize it on a regular basis. This equates to 63.2% of the global population having access to the Internet. According to statistics, Internet usage surged by 1266% over the past two decades [2,3]. The explosiveness and widespread nature of the Internet have made almost everyone rely on computer networks for their day-to-day activities [4]. With an immense rise in dependency on the Internet and computer networks services, attacks and malicious behaviors have become unexceptional in our computing environment [5–7].

The emergence of attacks and malicious behaviors pose a significant danger to computer security [8]. They attempt to deviate from the deployed network security mechanism by exploiting the vulnerabilities found in the target networks [4,6]. Computer system attacks are achievable at several levels, ranging from data link layer to application layer. Attacks can also be classified as passive or active attacks [9,10]. An active attack occurs when attackers change system resources and cause effect to their operations. A passive attack occurs when attackers gather or make use of information from the systems but do

not affect system resources [11,12]. Password-based attacks, like dictionary-based attacks and brute-force attacks, are among various types of computer attacks [9,13].

The brute-force attack, often referred to as high-level attack, is one among the most popular insurmountable challenges in today's computer system attacks [6,14–16]. In brute-force attack, attackers attempt to log in by trying different passwords on the victim's machine to reveal the login passwords [6,16–18]. They generate password combinations using automated tools. There are several smart brute-force attack tools available, including Hydra, the most well-known brute-force attack tool, which comes pre-installed in the Kali Linux operating system [6,16]. Brute-force attacks can be used against a wide range of services or protocols with SSH and FTP being among the primary targets for the attack.

In order to achieve dictionary-based or brute-force attack, an attacker needs to have two important items: *a valid and existing list of usernames of the targeted system* and *a wordlist dictionary* (a text file containing a collection of words for use in the attacks). One of the keys first steps when attempting to gain access or to launch an attack to a victim system or application is to enumerate usernames. This means an attacker first gathers the fundamental information about a user [19]. Once intended usernames have been enumerated, targeted password-based attacks can be launched against found usernames.

Username enumeration is a sort of a passive attack (reconnaissance) that retrieves a list of existing and valid usernames from a system that requires user authentication [20,21]. Since an attacker can quickly generate a list of legitimate usernames from the username enumeration attack, the time and effort necessary to brute-force a login is considerably reduced [22]. However, it does not allow the attacker to immediately log in, rather it gives half of the necessary information which the attacker could use to run a brute-force attack to further exploit the obtained information.

The username enumeration attacks can be initiated in any system that requires user authentication including, SSH servers. Specific versions of OpenSSH experience suffering from a timing-based attack: if a valid username with a long password is given, the time taken to respond is noticeably longer than for an invalid username with a long password [23]. By exploiting how the server responds to forged queries, the attacker can enumerate the service's registered usernames. The server would respond with an authentication failure if the username does not exist, but the outcome would be different if the user exists. Other areas where username enumeration occurs are in a website login page and its *'forgot password'* functionality.

The demand for traffic anomaly detection in cybersecurity is increasing because of the enormous and rapid expansion of computer attacks that are sophisticated, including password-based attacks [6]. Several approaches for detecting and mitigating password-related attacks, such as brute-force, have been suggested, developed, and deployed on a variety of systems and services, including SSH, FTP, and HTTP. However, in the era of cybersecurity, username enumeration attacks continue to be a problem. The majority of the recommended solutions focus on detecting and preventing password-based attacks, ignoring the fact that username enumeration is the first attack to identify and resist.

Inspired by the advancement and promising results of machine-learning techniques in traffic anomaly detection and mitigation [24–26], this study focuses on detection of the username enumeration attack on SSH protocol by applying and analyzing machine-learning classifiers.

Machine-learning is a branch of artificial intelligence that allows machines to learn without having to be plainly programmed [27]. Machine-learning automates operations by skillfully taking each stage in a maintained way. Machine-learning contains several learning techniques categorized as supervised and unsupervised learning. This categorization is subjected to the existence or nonexistence of labelled dataset. Supervised learning uses labelled samples to train the model, allowing it to anticipate comparable unlabeled samples. There are no training samples in unsupervised learning, hence it relies on the arithmetical method of density approximation. Unsupervised learning is based on the

notion of gathering or grouping data of the same types to uncover the underlying design of the data.

Machine-learning ability to recognize and give clues on real life issues is greatly valued and thus lead to their appeal and perverseness. These accomplishments have steered to the adoption of machine-learning in numerous fields [28,29]. Cybersecurity is among other fields availed by this trend where intrusion detection systems (IDS) are advanced with machine-learning modules [30]. With their real-time response and adaptive learning process, machine learning algorithms are becoming particularly efficient in intrusion detection systems [31]. They exemplify supreme choice over conventional rule-based algorithms [32].

Attacks and anomaly detection use supervised learning where a known dataset is used to make classification or prediction. The training dataset contains input features and target values. The supervised learning algorithm then builds a model to make classification or prediction of the target values [33].

In this work, we examine four machine-learning classifiers for the username enumeration attacks detection. We examine k-nearest-neighbor, naïve Bayes, random forest and decision tree machine-learning classifiers. The use of several classifiers offers a wider investigation spectrum of the machine-learners' ability in the detection of username enumeration attacks. Section III has more information on these classifiers.

Our findings show that utilizing machine-learning algorithms to detect SSH username enumeration attacks is a very successful approach. Additionally, we examine the impact of source and destination ports usage in the detection of username enumeration attacks. This is achieved by including source and destination ports as feature sets in model development and evaluation.

The remaining part of the paper is arranged out as follows: Section 2 discusses the works related to brute-force attacks and various detection methods. The experimental setup, dataset and dataset pre-processing, the classifiers we used are all presented in Section 3. We discuss our findings in Section 4. Finally, in Section 5, we wrap up our research and make recommendations for future investigation.

2. Related Works

The username enumeration attack to get a list of existing usernames works hand in hand with password-related attacks like brute-force. A typical brute-force attack looks for the right user and password combination, frequently without knowing if the user already exists on the system. The Verizon 2020 data breach investigation report highlighted that brute-force attacks accounted for more than 80% of all data breaches. It is a long-standing strategy, yet it is still prevalent and effective among hackers today [34]. In various research, the dominance of brute-force attack has indeed been observed.

One of the studies observed the prevalence of brute-force attack is [35], they examined the attack pattern on SSH protocol by investigating aggregated NetFlow data using decision tree classifier. Their study evaluation was conducted in a high-speed university campus network. Satoh et al. [36] investigated SSH dictionary attack by means of machine-learners. They subsequently suggested two novel elements for dictionary attack detection. The two studies had promising results, however, none of them ever addressed the issue of username enumeration attack.

Mobin et al. [37] studied distributed SSH brute-force attack detection by using statistical analysis on thousands of users' dataset collected for 8 years. They suggested that significant statistical changes in a parameter that summarizes aggregate activity revealed brute-force attack. They further indicated there is complexity implementation to some of the approaches for detecting specific attacks. In paper [6], the authors explored the detection of brute-force attack on SSH using NetFlow data examination under four machine-learning classifiers using their own generated labeled dataset. The two approaches proved to be successful with promising results. The focus was on detection of password-based attacks but there was no effort on detecting username enumeration attacks.

Kim et al. [38] investigated intrusion detection using KDDCUP99 dataset under LSTM recurrent neural network classifier and machine-learning algorithms. They afterward performed comparison of neural network results to machine-learning results and concluded the former outperformed the latter. Hossain et al. [16] also studied SSH and FTP brute-force attacks detection using LSTM and machine-learning classifiers. They also concluded that deep learning results outperformed machine-learning results. Similarly, both studies attained outstanding results, but none put focus on detecting the username enumeration attacks.

Hofstede et al. [39] delved into brute-force attacks on web applications and discussed several phases brute-force attacks undergo. They concluded that at a high-speed network, it is challenging to detect the attacks. Hynek et al. [40] proposed a study on redefined brute-force attack detection using a machine-learning approach. They used extended IP flow features obtained from backbone network traffic dataset to differentiate successful and unsuccessful login. Other research, in addition to the studies mentioned above, suggests that brute-force attacks are still amongst the most common attacks on the Internet [41].

All the aforementioned studies have focused and achieved excellent results on detecting and mitigating password related attacks such as brute force that are generated by various password attack tools. However, none of them have adequately included and addressed the issue of detection and mitigation of the username enumeration attacks. Considering that for any password-based attack to be launched, an attacker must have gathered all information including the list of usernames of the targeted system obtained from the username enumeration attack. Therefore, the detection and prevention of the username enumeration attack is highly needed in order to deny an opportunity for an attacker to retrieve a valid and existing list of usernames of the targeted system.

3. Materials and Methods

This section contains the following information: Experimental setup and attack scenario are explained in the first part. In the second part, network traffic data from a closed-environment network is collected and given corresponding labels, resulting in a new dataset. Third, several data pre-processing techniques are conducted in order to transform raw dataset into readable and understandable format by machine learning algorithms. As previously stated, the four classifiers are utilized to create classification models from the labeled traffic data. We carry out two-fold of experimentations seeing how using and not using ports information affects username enumeration attack detection. The rest of this section delves deeper into the steps listed above.

3.1. Experimental Setup

The attack simulation is carried out in a closed-environment network consisted of a victim machine, penetration testing platform and data collection point. The victim machine—SSH server was registered with thousands of users. The SSH server was a patched version of OpenSSH server version 7.7 [42] that listens on standard TCP port 22 for incoming and outgoing traffic. We chose this version because the attack occurs between version 2.3 and 7.7 [43]. The SSH server runs on Ubuntu Linux 20.04 (×64) with a 2.8 GHz Intel Core i7 CPU and a 16GB RAM computer. A penetration testing platform—Kali Linux 2020.4 (×64) with kernel version 5.9.0—is targeting this SSH server. This penetration platform operates on a machine with a 16 GB of RAM and 3.4 GHz Intel Core i7 CPU. The data collection server runs on Linux Mint 20.2 with 16 GB RAM computer, 2.8 GHz Intel Core i7 CPU. The IP addresses for the SSH server, penetration testing system and data collection server are 192.168.56.115, 192.168.100.117, 192.168.100.16 respectively, and are in the private IPv4 range.

3.2. Attack Scenario

The attack was launched from Kali Linux, a penetration testing platform, to SSH server, a victim machine. The common vulnerabilities and exposures (CVE) with the identification number CVE-2018-15473 retrieved from the public exploits database [43] were used to

do this. The CVE is developed entirely in Python language. The CVE mentioned above generates username enumeration attack traffic from the penetration testing platform, Kali machine, to a victim machine, SSH server. The attack was accomplished by employing the attack command shown in Figure 1.

```
root@kali:~# python CVE-2018-15473.py --userList ssh_users.txt --outputFile validusers.
txt 192.168.56.115
[+] Results successfully written to validusers.txt in List form.
root@kali:~#
```

Figure 1. Username enumeration command.

Figure 2 depicts the attack's output by listing all the usernames found on the SSH server, including the root account. It displays a list of all existing usernames by indicating *"valid user"* and *"is not a valid user"* for those not found in the system. To get a mix of normal and attack traffic, a *pcap* file of normal traffic was obtained from public training repository [44]. The pcap file was replayed by using *tcpreplay* [45] tool at the same time when an attack was launched from Kali machine to the SSH server. Finally, both traffic, attack and normal, were collected in data collection point.

```
admin is not a valid user!
root is a valid user!
sys is a valid user!
auditor is not a valid user!
abel is a valid user!
flo is not a valid user!
emmy is not a valid user!
1234jeje is not a valid user!
root@kali:~#
```

Figure 2. Output of username enumeration.

3.3. Data Collection and Labelling

The dataset is collected from a closed-environment network using network monitoring tools *tcpdump* [46] and *Wireshark* [47] installed in the data collection point. A total of 36,273 raw packet data were collected, each containing 25 features with label exclusive. The packet data were then given their corresponding labels as username enumeration attack and non-username enumeration attack. We chose the terms *"username enumeration attack"* and *"non-username enumeration"* instead of the traditional *"attack"* and *"normal"* label notations since *"normal"* traffic data could contain attacks other than username enumeration attack, which is the focus of our research. Since the goal of this study is to detect username enumeration attacks, we found that labeling dataset in this way is more suitable. The username enumeration attack class corresponds to the attack traffic while non-username enumeration class corresponds to the normal traffic. This traffic reflects different services including emails, DNS, HTTP, web, few to mention. We finally managed to get a raw dataset [48] comprising attack traffic and normal traffic. The dataset was then split into a training subset and a testing subset with an 80/20 ratio to deliver evaluation results on the classifiers' efficacy. The dataset split was based on Pareto Principle [49], also known as 80–20 rule. The 80–20 split ratio is indicated as one of the most common ratios in the machine learning and deep learning fields and was used in similar work in intrusion detection systems such as [16]. The distribution of the dataset is indicated in Tables 1 and 2.

Table 1. Dataset collected.

Class	Instances in Each Class
SSH username enumeration attack	18,844
Non-username enumeration	17,429
Total instances	36,273

Table 2. Dataset splitting.

Class	Instances	Training Set	Testing Set
Username enumeration	18,844	15,075	3769
Non-username enumeration	17,429	13,943	3486

3.4. Data Preprocessing

The Data pre-processing is the data mining technique that transforms raw datasets into readable and understandable format. Machine learning algorithms make use of the datasets in mathematical format, such format is achieved through data pre-processing [50]. Among other techniques of data pre-processing include missing-data treatment, categorical encoding, data projection and data reduction. Missing-data treatment involves deletion of missing values or replacement with estimations. Categorical encoding aims to transform categorical values into numerical values. Data projection scales the values into a symmetric range and this helps to change the appearance of the data. Data reduction intends to reduce the size of datasets using several techniques including features selection.

In this work, the missing values in a dataset were treated using imputation technique. For the categorical features, the *most frequent* strategy was used within each column. For the case of numerical features, a *constant* strategy was implemented to replace the missing values. Both label encoding and one hot encoding techniques were used to transform categorical feature values into numerical feature values. Hence, two types of datasets were generated. However, in this work label encoding dataset was used. Though one hot encoding is a common method, it faces a challenge of increasing the dimension of the dataset contrary to the label encoding approach which straightly converts the nominal feature values into specific numerical feature values. All features were scaled into the predefined same range using *MinMaxScaler()* method. Dataset reduction was implemented using features selection method. We selected 7 different features from the dataset. The description of each feature is shown in Table 3. All the data pre-processing techniques were carried out using scikit-learn library.

Table 3. Description of features selected.

Feature Name	Feature Description
Time	Packet duration time in seconds
Packet Length	The length of the packet in bytes
Delta	Time interval between packets in seconds
Flags	Flags seen in the packet
Total Length	The total length of the packet in bytes
Source Port	The source port of the packet
Destination Port	The destination port of the packet

3.5. Applying Machine-Learning Classifiers to Dataset

In this work, we picked four distinctive machine-learning classifiers for our study. We examine k-nearest- neighbor, naïve Bayes, random forest and decision Tree machine-learning classifiers. We picked different classifiers to investigate a wider scale of investigation in username enumeration attack detection. These classifiers have asymmetric features and have light weight computation. A brief explanation for each classifier picked is provided below. We developed all models using scikit-learn library under GPU environment using python v3.7. All the models were built by tuning their parameters. Table 4 shows parameters tuning for each model.

Table 4. Hyperparameter used for model training.

Classifier	Hyperparameter	Value
Random Forest (RF)	Bootstrap	True
	Maximum depth	90
	Maximum features	Auto
	Minimum sample leaf	1
	Minimum sample split	5
	N estimators	1600
Decision Tree (DT)	Criterion	Gini
	Maximum depth	50
	Maximum features	Auto
	Maximum leaf nodes	950
	Splitter	Best
Naïve Bayes (NB)	Var._Smoothing	$2.848035868435799 \times 10^{-5}$
K-Nearest Neighbors (KNN)	N	4
	Leaf size	7
	P	1

A decision tree is a widely known machine-learning classifier created in a tree-like structure [51]. It contains the internal nodes which represent attributes and branches and leaf nodes which represent the class label. To form classification rules, the root node is firstly selected which is a notable attribute for data separation. The path is then chosen from the root node to the leaf node. Decision tree classifier operates by recognizing associated attribute values as input data and produces decisions as output [52].

Random Forest is another dominant machine-learning classifier under the category of supervised learning algorithms [53]. Similarly, random forest is also used in machine-learning classification problems. This classifier is conducted in two asymmetric steps. The first step creates the asymmetrical forest of the specified dataset and the second one makes the prediction from the classifier acquired in the initial stage [54].

Naïve Bayes is a common probabilistic machine-learning classifier used in classification or prediction problems. It operates by calculating the probability to classify or predict a certain class in a specified dataset. It contains two probabilities: class and conditional probabilities. Class probability is the ratio of every class instance occurrence to the total instances. Conditional probability is the quotient of every feature occurrence for a certain class to the sample occurrence of that class [55,56]. Naïve Bayes classifier presumes every attribute as asymmetry and contemplates association between the attributes [57].

K-Nearest Neighbors is a classifier that considers three important elements in its classification manner: record set, distance, and value of K [58]. It functions by calculating the distance between sample points and training points. The smallest distance point is the nearest neighbor [59]. The nearest neighbor is measured with respect to the value of k (in our case k = 4), this defines the number of nearest neighbors required to be examined in order to define the class of sample data point [60].

We built all four classification models using a subset of 80% data of the given dataset and used the remaining subset of 20% for testing the models. The train test split ratio for each classifier was even. The performance metrics to evaluate the effectiveness of our de-veloped models were computed in terms of precision, recall and overall accuracy. The metrics are defined below.

$$\text{Precision} = \text{True Positive}/(\text{True Positive} + \text{False Positive}) \quad (1)$$

$$\text{Recall} = \text{True Positive}/(\text{True Positive} + \text{False Negative}) \quad (2)$$

The Receiver Operating Characteristics (ROC) curve was also considered as an additional performance metric. This evaluation metric draws the graph of True Positive against the False Positive of the subsequent model. It shows the difference amid True Positive rate

and False Positive rate where the higher ROC value indicates high True Positive rate and low False Positive rate which is desirable in anomaly detection.

We conducted two types of experimentations; one excludes source and destination ports and the other includes them as our input features. This is because sometimes network administrators do customize the destination port to some different number other than the default port number for SSH protocol which is port 22. With these two experiments, we observed that including and excluding ports information has significant impact on the classification outcomes. The outcome scores advocate that using ports information as input features improves performance metrics of the developed models based on the kind of classifier used. However, excluding port information as input features in the dataset also provides significant benefits of developing a sturdy model that portrays the situation when SSH protocol is not configured in the standard default port.

4. Results and Discussion

For each classification model developed, we used the same training set and test set. 80% data of the given dataset was used for training the classification models and the rest 20% data was used to test the models. Tables 5 and 6 show the results of four developed machine-learning based classification models when port information is included and not included as a feature set.

Table 5. Performance metrics—Ports exclusive.

Classifier	Precision	Accuracy	ROC
DT	99.84	99.88	0.997
RF	99.87	99.92	0.998
NB	94.85	95.70	0.994
KNN	99.95	99.93	0.999

Table 6. Performance metrics—Ports inclusive.

Classifier	Precision	Accuracy	ROC
DT	99.97	99.93	0.998
RF	99.89	99.94	0.999
NB	99.72	99.85	0.997
KNN	100	99.95	1.000

If we observe our prediction results, we see all the classification models in both tables—when including and excluding ports information provide outstanding results as indicated by an accuracy of greater than 95.70%, that ensures the models effectiveness in the detection of username enumeration attack. The KNN classifier has the maximum performance metrics with an accuracy of 99.95% when including source and destination ports as input features and an accuracy of 99.93% while excluding source and destination ports as models input features.

Additionally, Figures 3 and 4 show the ROC curves as the models' outcome results for two kinds of experiments conducted. They represent the True positive rate versus False Positive rate of each classification model developed. From the figures, we observe that the correctly classified rate is higher close to the maximum value of 1 while the falsely classified rate is low for both cases—when including and excluding ports information. Therefore, from the outcome results in Tables 5 and 6 together with ROC curves in Figures 3 and 4, we can conclude that our machine-learning based classification models are effectively able to detect username enumeration attack with high detection rate and low false alarm rate.

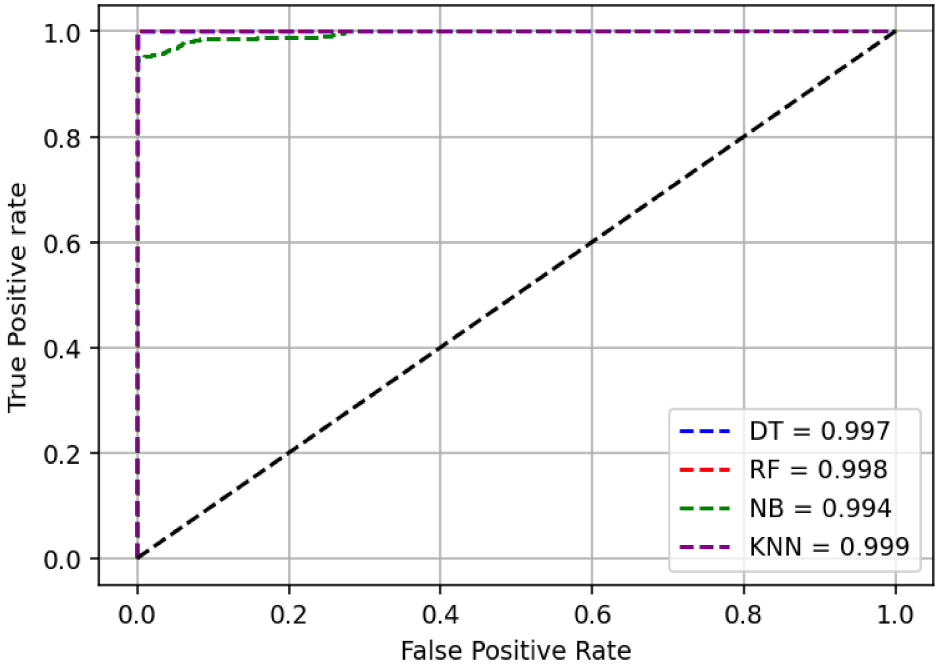

Figure 3. ROC AUC—Ports Exclusive.

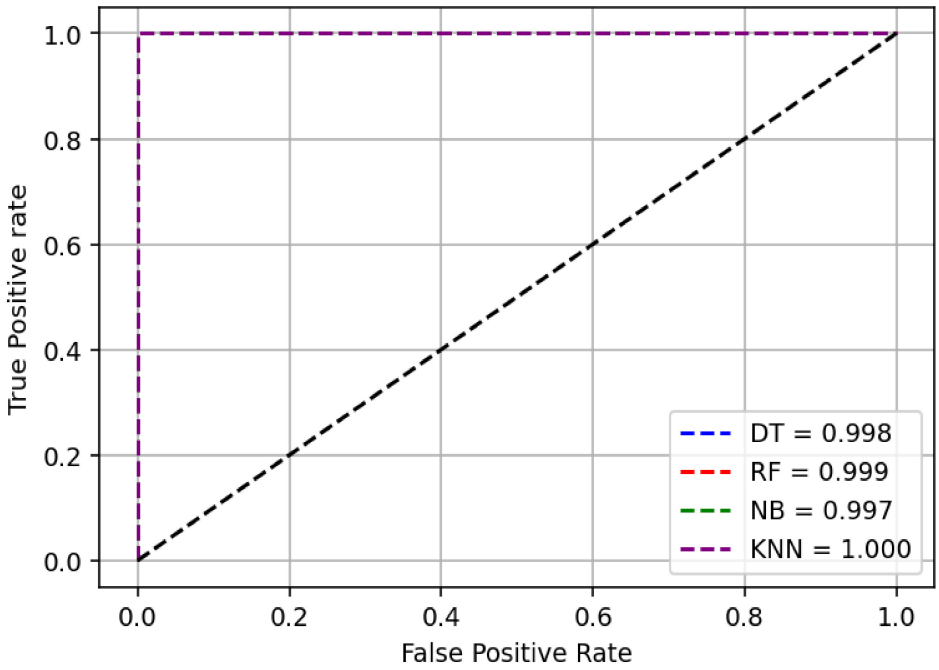

Figure 4. ROC AUC- Ports inclusive.

Effectiveness Comparison When Including and Excluding Ports Information

The effectiveness comparison between two kinds of experiments conducted shows that when including source and destination ports as input features, there are performance improvements compared to when source and destination ports are excluded as input features. Tables 5 and 6 show the relative comparison of precision, accuracy and roc-auc utilizing the dataset discussed in the earlier section.

The classification performances of the DT, RF, and KNN models slightly improve. KNN model increases from an accuracy of 99.93% when excludes source and destination ports as feature set to an accuracy of 99.95% when includes source and destination as feature set. Similarly, the RF model slightly improves from an accuracy of 99.92% to 99.94%

when including source and destination port as the model's input features. The decision tree improves its performance from an accuracy of 99.88% to 99.93%. The naïve Bayes model has a significant improvement when including ports information as a feature set. It increases from an accuracy of 95.70% to 99.85%. Usually, naïve Bayes is a weak classifier and for the case of excluding ports information as input features in our study, other classifiers outperform it. However, by including source and destination port to its feature set naïve Bayes produces almost the same performance outcome results compared to DT, RF and KNN.

We observe that the DT, RF and KNN classification models produce almost the same classification performances regardless of whether port information is included or excluded in the feature set. This can be translated that even if source and destination ports are not included as model's input features, the distribution of samples in the feature area is still a means that samples with the symmetry label are dispersed together.

We also observe that naïve Bayes classification model has a significant enhancement of performance when including ports information as its input feature. This is due to the presumption that features in naïve Bayes are completely independent. Therefore, it is ra-tional to accept that the independency nature of naïve Bayes' features can be recompensed with inclusion of additional attributes to its attribute set and yields in performance improvement.

Thus, according to the results shown in Tables 5 and 6 and the above experimental analysis, we can conclude that including source and destination ports as input features has various impacts on the developed classifiers depending on their type; however, generally it enhances the performances, ensuring the models' effectiveness in the detection of the username enumeration attacks.

5. Conclusions

In this paper, we present a novel SSH username enumeration attack detection method using machine-learning approaches. To achieve this, we collected the data from a closed-environment network and the dataset is then labelled to generate a labelled dataset. We trained four distinct classifiers in a dataset containing username enumeration and non-username enumeration attack class instances. The former represented the normal class while the latter represented the attack class. We evaluated the models' performance using accuracy, precision, and ROC-AUC values. Our findings show that, using machine-learning approaches in detecting SSH username enumeration attacks, we can achieve reasonable results with KNN having an accuracy of 99.93%, NB 95.70%, RF 99.92%, and DT 99.88%.

In addition, when training classification models, we investigated the impact of including ports information in the feature set. Our findings imply that, including source and destination ports as input features resulted in some performance improvements without compromising computation power. However, the performance improvements vary from classifier to classifier based on their nature. Naïve Bayes has a significant enhancement of performance when including ports information. Naïve Bayes' features are completely independent, hence, including ports information yields significant performance improvements.

In the future work, we aim at gathering data in a production-environment network and evaluate how developed models would perform on the real-world live dataset. Deep-learning techniques may also be incorporated in the future to detect username enumeration attacks.

Author Contributions: Literature review, A.Z.A.; conceptualization, A.Z.A. and J.D.N.; methodology, A.Z.A., L.J.M. and J.D.N.; writing-original draft, A.Z.A.; validation, L.J.M., S.M.P. and M.A.D.; writing—review and editing, J.D.N.; co-supervision, S.M.P. and M.A.D.; supervision, J.D.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the novelty of the study, the dataset had to be generated through the use of public exploits and pcap files from public training repositories. The generated datasets

are publicly available to everyone and can be found at https://doi.org/10.5281/zenodo.5564663 (accessed on 9 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alshehri, H.; Meziane, F. Current state on internet growth and usage in Saudi Arabia and its ability to support e-commerce development. *J. Adv. Manag. Sci.* **2017**, *5*, 127–132. [CrossRef]
2. Infante-Moro, A.; Infante-Moro, J.-C.; Martínez-López, F.-J.; García-Ordaz, M. The importance of internet and online social networks in the Spanish hotel sector. *Appl. Comput. Sci.* **2016**, *12*, 75–86.
3. World Internet Users Statistics and 2021 World Population Stats. 2021. Available online: https://www.internetworldstats.com/stats.htm (accessed on 21 May 2021).
4. Hoque, N.; Bhuyan, M.H.; Baishya, R.C.; Bhattacharyya, D.K.; Kalita, J.K. Network attacks: Taxonomy, tools and systems. *J. Netw. Comput. Appl.* **2014**, *40*, 307–324. [CrossRef]
5. Jaw, E.; Wang, X. Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry* **2021**, *13*, 1764. [CrossRef]
6. Najafabadi, M.M.; Khoshgoftaar, T.M.; Kemp, C.; Seliya, N.; Zuech, R. Machine learning for detecting brute force attacks at the network level. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Bioengineering, Boca Raton, FL, USA, 10–12 November 2014; pp. 379–385.
7. Jang-Jaccard, J.; Nepal, S. A survey of emerging threats in cybersecurity. *J. Comput. Syst. Sci.* **2014**, *80*, 973–993. [CrossRef]
8. Meryem, A.; Ouahidi, B.E.L. Hybrid intrusion detection system using machine learning. *Netw. Secur.* **2020**, *2020*, 8–19. [CrossRef]
9. Pawar, M.V.; Anuradha, J. Network security and types of attacks in network. *Procedia Comput. Sci.* **2015**, *48*, 503–506. [CrossRef]
10. Sheikh, A.F. *CompTIA Security+ Certification Study Guide*; Apress: Berkeley, CA, USA, 2020. [CrossRef]
11. Liu, Y.; Morgan, Y. Security against passive attacks on network coding system—A survey. *Comput. Netw.* **2018**, *138*, 57–76. [CrossRef]
12. Srivastava, M. An Introduction to Network Security Attacks. In *Inventive Systems and Control*; Springer Nature: Singapore, 2021; pp. 505–515. [CrossRef]
13. Nagamalai, D.; Renault, E.; Dhanuskodi, M. *Trends in Computer Science, Engineering and Information Technology: Proceedings of the First International Conference (CCSEIT) Tirunelveli, Tamil Nadu, India, 23–25 September 2011*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 204.
14. Alata, E.; Nicomette, V.; Kaâniche, M.; Dacier, M.; Herrb, M. Lessons learned from the deployment of a high-interaction honeypot. In Proceedings of the Sixth European Dependable Computing Conference, Coimbra, Portugal, 18–20 October 2006; pp. 39–46.
15. Hewlett-Packard Development Company. Top Cyber Security Risks Threat Report for (2010). Available online: http://dvlabs.tippingpoint.com/toprisks2010 (accessed on 4 June 2021).
16. Hossain, M.D.; Ochiai, H.; Doudou, F.; Kadobayashi, Y. SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches. In Proceedings of the 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 22–24 February 2020; pp. 491–497.
17. Anandita, S.; Rosmansyah, Y.; Dabarsyah, B.; Choi, J.U. Implementation of dendritic cell algorithm as an anomaly detection method for port scanning attack. In Proceedings of the 2nd International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 16–19 November 2015; pp. 1–6.
18. Vykopal, J. A flow-level taxonomy and prevalence of brute force attacks. In Proceedings of the International Conference on Advances in Computing and Communications (ACC), Kochi, India, 22–24 July 2011; pp. 666–675.
19. Dave, K.T. Brute-force Attack 'Seeking but Distressing'. *Int. J. Innov. Eng. Technol. Brute Force* **2013**, *2*, 75–78.
20. Li, P.; Qiu, X. NodeRank: An algorithm to assess state enumeration attack graphs. In Proceedings of the 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 21–23 September 2012; pp. 1–5.
21. Virtue Security. *Username Enumeration*. 2021. Available online: https://www.virtuesecurity.com/kb/username-enumeration/ (accessed on 28 June 2021).
22. Portswigger—Web Security Academy. 2018. Vulnerabilities in Password-Based Login. Available online: https://portswigger.net/web-security/authentication/password-based (accessed on 22 April 2021).
23. Kannisto, J.; Harju, J. The time will tell on you: Exploring information leaks in ssh public key authentication. In Proceedings of the 11th International Conference on Network and System Security, Helsinki, Finland, 21–23 August 2017; pp. 301–314.
24. Elmrabit, N.; Zhou, F.; Li, F.; Zhou, H. Evaluation of machine learning algorithms for anomaly detection. In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, 15–17 June 2020; pp. 1–8.
25. Eltanbouly, S.; Bashendy, M.; AlNaimi, N.; Chkirbene, Z.; Erbad, A. Machine learning techniques for network anomaly detection: A survey. In Proceedings of the IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 156–162.
26. Nawir, M.; Amir, A.; Yaakob, N.; Lynn, O.B. Effective and efficient network anomaly detection system using machine learning algorithm. *Bull. Electr. Eng. Inform.* **2019**, *8*, 46–51. [CrossRef]

27. Mahesh, B. Machine Learning Algorithms—Review Self Flowing Generator View Project Machine Learning Algorithms. *Int. J. Sci. Res.* **2020**, *9*, 381–386. [CrossRef]
28. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In Proceedings of the 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May–1 June 2018; pp. 371–390.
29. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
30. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
31. Ahsan, M.; Gomes, R.; Chowdhury, M.; Nygard, K.E. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. *J. Cybersecur. Priv.* **2021**, *1*, 199–218. [CrossRef]
32. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
33. Ndibwile, J.D.; Govardhan, A.; Okada, K.; Kadobayashi, Y. Web server protection against application layer DDoS attacks using machine learning and traffic authentication. In Proceedings of the IEEE 39th Annual Computer Software and Applications Conference, Taichung, Taiwan, 1–5 July 2015; Volume 3, pp. 261–267. [CrossRef]
34. Nathan, A.J.; Scobell, A. 2020 Data Breach Investigations Report. *Verizon*. 2020. Available online: https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf%0Ahttp://bfy.tw/HJvH (accessed on 12 July 2021).
35. Vykopal, J.; Plesnik, T.; Minarik, P. Network-based dictionary attack detection. In Proceedings of the International Conference on Future Networks, Bangkok, Thailand, 7–9 March 2009; pp. 23–27.
36. Satoh, A.; Nakamura, Y.; Ikenaga, T. SSH dictionary attack detection based on flow analysis. In Proceedings of the IEEE/IPSJ 12th International Symposium on Applications and the Internet, Izmir, Turkey, 16–20 July 2012; pp. 51–59.
37. Javed, M.; Paxson, V. Detecting stealthy, distributed SSH brute-forcing. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 85–96.
38. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016; pp. 1–5.
39. Hofstede, R.; Jonker, M.; Sperotto, A.; Pras, A. Flow-based web application brute-force attack and compromise detection. *J. Netw. Syst. Manag.* **2017**, *25*, 735–758. [CrossRef]
40. Hynek, K.; Beneš, T.; Čejka, T.; Kubátová, H. Refined Detection of SSH Brute-Force Attackers Using Machine Learning. In Proceedings of the 35th IFIP International Conference on ICT Systems Security and Privacy Protection, Maribor, Slovenia, 21–23 September 2020; pp. 49–63.
41. Stiawan, D.; Idris, M.; Malik, R.F.; Nurmaini, S.; Alsharif, N.; Budiarto, R. Investigating Brute Force Attack Patterns in IoT Network. *J. Electr. Comput. Eng.* **2019**, *2019*, 4568368. [CrossRef]
42. OpenSSH. 2021. Available online: https://www.openssh.com/ (accessed on 18 August 2021).
43. Exploit Database. *OpenSSH 2.3 < 7.7—Username Enumeration*. 2018. Available online: https://www.exploit-db.com/exploits/45233 (accessed on 21 August 2021).
44. Stratosphere Lab. *Malware Capture Facility Project: Normal Captures—Stratosphere IPS*. 2019. Available online: https://www.stratosphereips.org/datasets-normal (accessed on 21 August 2021).
45. Li, Y.; Miao, R.; Alizadeh, M.; Yu, M. {DETER}: Deterministic {TCP} Replay for Performance Diagnosis. In Proceedings of the 16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19), Boston, MA, USA, 26–28 February 2019; pp. 437–452.
46. TCPDUMP/LIBPCAP Public Repository. 2021. Available online: https://www.tcpdump.org/ (accessed on 5 September 2021).
47. Wireshark. 2021. Available online: https://www.wireshark.org/ (accessed on 5 September 2021).
48. Agghey, A. SSH Username Enumeration Attack Detection Dataset. *Zenodo* **2021**. [CrossRef]
49. Dunford, R.; Su, Q.; Tamang, E. The pareto principle. *Plymouth Stud. Sci.* **2014**, *7*, 140–148.
50. Huang, J.; Li, Y.F.; Xie, M. An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Inf. Softw. Technol.* **2015**, *67*, 108–127. [CrossRef]
51. Cherfi, A.; Nouira, K.; Ferchichi, A. Very fast C4. 5 decision tree algorithm. *Appl. Artif. Intell.* **2018**, *32*, 119–137. [CrossRef]
52. Yang, F.J. An extended idea about decision trees. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NA, USA, 5–7 December 2019; pp. 349–354. [CrossRef]
53. Li, X.; Chen, W.; Zhang, Q.; Wu, L. Building auto-encoder intrusion detection system based on random forest feature selection. *Comput. Secur.* **2020**, *95*, 101851.
54. Bhavani, T.T.; Rao, M.K.; Reddy, A.M. Network intrusion detection system using random forest and decision tree machine learning techniques. In Proceedings of the 1st International Conference on Sustainable Technologies for Computational Intelligence, Jaipur, India, 29–30 March 2019; pp. 637–643.
55. Alqahtani, H.; Sarker, I.H.; Kalim, A.; Hossain, S.M.M.; Ikhlaq, S.; Hossain, S. Cyber intrusion detection using machine learning classification techniques. In Proceedings of the International Conference on Computing Science, Communication and Security, Gujarat, India, 26–27 March 2020; pp. 121–131.
56. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. *arXiv* **2013**, arXiv:1302.4964. Available online: https://arxiv.org/abs/1302.4964v1 (accessed on 16 August 2021).

57. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Morgan Kaufmann Publishers: Waltham, MA, USA, 2011.
58. Malhotra, S.; Bali, V.; Paliwal, K.K. Genetic programming and K-nearest neighbour classifier based intrusion detection model. In Proceedings of the 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, Noida, India, 12–13 January 2017; pp. 42–46.
59. Bhatia, N. Survey of Nearest Neighbor Techniques. *arXiv* **2010**, arXiv:1007.0085. Available online: https://arxiv.org/abs/1007.0085v1 (accessed on 17 August 2021).
60. Soofi, A.A.; Awan, A. Classification techniques in machine learning: Applications and issues. *J. Basic Appl. Sci.* **2017**, *13*, 459–465. [CrossRef]