Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 02, February : 2022

# Exploring Intrusion Detection Systems: A Survey Using Cloud Intrusion Dataset

Pratyush Ranjan Mohapatra,Kamalakanta Shaw,Abhilash Pati
Dept. of Computer Science and Engineering,Gandhi Institute For Technology,Bhubaneswar,752054
Email: pratyush@gift.edu.in

**ABSTRACT**

Nowadays society, economy, and critical infrastructures have become principally dependent on computers, networks, and information technology solutions, on the other side, cyber-attacks are becoming more sophisticated and thus presenting increasing challenges in accurately detecting intrusions. Failure to prevent intrusions could compromise data integrity, confidentiality, and availability. Different detection methods are proposed to tackle computer security threats, which can be broadly classified into anomaly-based intrusion detection systems (AIDS) and signature-based intrusion detection systems (SIDS). One of the most preferred AIDS mechanisms is the machine learning-based approach which provides the most relevant results ever, but it still suffers from disadvantages like unrepresentative dataset, indeed, most of them were collected during a limited period of time, in some specific networks and mostly don't contain up-to-date data. Additionally, they are imbalanced and do not hold sufficient data for all types of attacks, especially new attack types. For this reason, up- to-date datasets such as information security and object technology-cloud intrusion dataset (ISOT-CID) are very convenient to train predictive models on a cloud-based intrusion detection approach. The dataset has been collected over a sufficiently long period and involves several hours of attack data, culminating into a few terabytes. It is large and diverse enough to accommodate machine-learning studies.

## 1. INTRODUCTION

In order to detect intrusion activity, various tools are used in industry as well as in research organizations, such as firewall, antivirus, and intrusion detection system (IDS). IDS, which have long been a topic for developing theoretical and research, are gaining mainstream popularity as structures move more of their critical business activities to the Internet. An intrusion detection system can be signature-based IDS when its processing consists of comparing the already known attacks with the incoming network traffic to detect the intrusions that are stored in the database as signatures. Existing attacks are well detected, but it often fails to detect novel attacks. The next category is called anomaly-based IDS. For this category, the normal traffic is modeled by the IDS models through learning patterns in the training phase, the deviations from these learned patterns are labeled as anomalies or intrusions. The implementation of real-time anomaly-based IDS is a colossal task because of the rapid increase in the network traffic behavior and relatively limited availability of computational resources (computation time and memory). Furthermore, these IDS need to be trained via a machine learning model by processing a representative dataset. Most of the works on this topic adopted old datasets, which contain out-to-date and redundant information and unbalanced volumes of data classes. The efficiency of an IDS is directly related to the selected learning model and the quality of the used dataset. A good quality dataset can be defined as a dataset that improves better performance metrics in real-world transactions. As mentioned in [1], imbalanced datasets present a problem to researchers. A dataset is said to be imbalanced when the distribution of classes of attack is not uniform. This is a common problem in many of the existing classifications. An imbalanced dataset results in the used classifier biases towards the majority class; however, in most of them, the aim is to try to detect the minority class [2]. Therefore, in this paper, we aim to use an up-to-date dataset (ISOT-CID) for training the IDS to develop a real knowledge base for the detection of an anomaly. We will precisely propose a new data distribution of the dataset to resolve the imbalance for the training volume (70% of the data volume) the remaining part is used for testing and validation. We experienced the following models: Logistic regression, decision tree, gradient boosting, Gaussian Naive Bayes, random forest, and artificial neural networks. The detailed mothology and results are exposed.

## 2. BACKGROUND AND RELATED WORK

A lot of research work has been carried out in intrusion detection system either in host-based intrusion detection (HIDS) or Network-based intrusion detection (NIDS), but there is no comprehensive reliable cyber dataset that covers both modern-day attacks and contemporary network intrusion detection system. The former being an individual device detecting a compromise and the latter detecting a compromise

in transit over a network [3]. NIDS can be further categorized into the anomaly and signature-based systems [4]. Signature-based systems form the mainstay of commercial network intrusion detection systems with anomaly-based still largely a research concept [5] with only a few practical vendors funded examples. Until today, there is a lot of academic research developed using machine learning supervised and unsupervised techniques [6]. Researchers are also using the combination of these techniques in recent years and they gain a high accuracy rate [7]. These results are discussable because the datasets which are used in the training phase are mostly out-of-date. Therefore, new attacks which have been discovered after creating the dataset cannot be imported to the anomaly database easily [8]. Researchers cannot decide whether these new attacks can be recognized or not in a real-time environment hence the importance of using an up-to-date dataset for such works.

## 3. ISOT-CID OVERVIEW

The information security and object technology (ISOT) cloud intrusion detection is provided by the ISOT research Lab which was founded in 1999 [9], and since then has been carrying innovative research in computer security and software engineering. The general approach adopted in this Lab consists of using knowledge from the fields of artificial intelligence and mathematics (e.g. formal logic, probability theory) to address some of the challenges posed by dependable and secure computing. The lab was especially aware that developing cloud-based IDS is very challenging as cloud IDS researchers are faced with one of the greatest hurdles: the lack of publicly available datasets collected from a real cloud-computing environment, which is a big hindrance for developing and testing realistic detection models. The ISOT-CID is one of the first public dataset of its kind collected from a production cloud environment as an initial response toward addressing such need and paving the way for cloud security communities for more research and findings. The dataset consists of over 2.5 terabytes of data, involving normal activities and a wide variety of attack vectors, collected in two phases (phase 1 in December 2016 and phase 2 in February 2018) [10] and over several months for the virtual machine instances, and several days and time slots for the hypervisors. The benign/normal data are from web applications and administrative activities ranging from maintaining the status of VMs, rebooting, updating, creating files, SSHing to the machines, and logging in a remote server. The web traffic was generated by more than 160 legitimate visitors, including more than 60 human users and genuine traffic generated by 100 robots, performing tasks such as account registration, reading/posting and commenting on blogs, browsing various pages, and so on. One of the web applications consists of a password management service for registered users. Labelling is provided for network traffic data in the form of comma-separated values (CSV) files which we used in this study where each row contains the header data (for identification) and the classification of a single packet. Each file contains all packets of the day specified by its name.

## 4. INCORPORATE MACHINE LEARNING CLASSIFIERS INTO ISOT-CID

### 4.1. Logistic regression classifier

In the next section we will present the different models that we studied with the ISOT-CID dataset; we will provide a brief overview of each one of them. The objective of logistic regression (LR) is to create the best suitable model to establish a relationship or dependence between the class variable and the features [11]. For a test case with only two classes: 0 and 1, it basically predicts a value between 0 and 1 which is the probability that the class is 1 for an observation. The simple LR model is only suitable for binary classification, but with effort can be extended for a multiclass purpose [12], [13].

### 4.2. Decision tree classifier

The perception behind the decision tree algorithm is simple, yet also very efficient. It requires relatively less effort for training the algorithm and can be used to classify non-linearly separable data. It is very fast and efficient compared to other classification algorithms. Entropy and information Gainare the most commonly used attribute selection measures for this type of classifier:

- Entropy: Entropy is the degree or amount of uncertainty in the randomness of elements or in other words it is a measure of impurity [14].
- Information Gain: It measures the relative change in entropy with respect to the independent attribute. It tries to estimate the information contained by each attribute. Creating a decision tree is all about finding the attribute that returns the highest information gain [15].

### 4.3. Gradient boosting classifier

Gradient boosting re-defines boosting as a numerical optimisation problem where the objective is to minimise the loss function of the model by adding weak learners using gradient descent. Gradient descent is a first-order iterative optimisation algorithm for finding a local minimum of a differentiable function [16]. As gradient boosting is based on minimising a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, and etcetera.

Gradient boosting does not modify the sample distribution as weak learners train on the remaining residual errors of a strong learner (i.e, pseudo-residuals) [17]. By training on the residuals of the model, this is an alternative resource to give more importance to misclassified observations. Intuitively, new weak learners are being added to concentrate on the areas where the existing learners are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimisation process to minimise the overall error of the strong learner.

### 4.4. Gaussian Naive Bayes classifier

**N**aive Bayes is a classification algorithm for two-class (binary) and multi-class classification problems. The technique is easier to understand when described using binary or categorical input values [18]. The class probabilities are simply the frequency of instances that belong to each class divided by the total number of instances. Moreover, the conditional probabilities are the frequency of each attribute value for a given class value divided by the frequency of instances with that class value. Training is fast because only the probability of each class and the probability of each class given different input values need to be calculated. No coefficients need to be fitted by optimization procedures.

### 4.5. Random forest classifier

Random forest is a supervised learning algorithm. It can be used for both classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees, it is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a good indicator of the feature importance [19].

### 4.6. Artificial neural network classifier

A neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer [20]. Generally, the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand [21]. This is the learning phase.

Neurons are organized into layers: input, hidden and output. The input layer is composed not of full

neurons, but rather consists simply of the record's values that are inputs to the next layer of neurons. The next layer is the hidden layer. Several hidden layers can exist in one neural network. The final layer is the output layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to the class node with the highest value [22].

## 5. IMPLEMENTATION

### 5.1. Artificial neural networks model

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of "classes.". To make our machine leaning classifiers, we used Anaconda and the most recent version of Numpy, Pandas and Sklearn. All packages are managed through Anaconda and pip, we also used Pycharm as our code editor.

### 5.2. Steps to incorporate the various machine learning classifiers

#### 5.2.1. Pre-processing of the dataset

The dataset is collected in two phases (phase 1 in December 2016 and phase 2 in February 2018), for each phase we construct a unique CSV file on wich we build our machine learning model, we particularily have carried out the following operations,

Encoding of non-numeric type columns as,

- 'Protocol' column: {"tcp": "0", "udp": "1"}
- 'flags' column: {"ACK": "0", "ACK CWR": "1", "ACK END": "2", "ACK END PSH": "3", "ACK END CWR": " 4 "," ACK PSH ":" 5 "," ACK PSH CWR ":" 6 "," ACK RST ":" 7 "," ACK RST CWR ":" 8 "," RST ":" 9 "," SYN ":" 10 "," SYN ACK ":" 11 "," SYN ACK ECE ":" 12 "," SYN ECE CWR ":" 13 "}
- 'fragement' column: {False: "0", True: "1"}
- 'Classification' column: {"benign": "1", "malicious": "0"}

#### 5.2.2. Correcting the imbalance ratio

According to the equation,

$$Imbalance\ ratio = \rho = \frac{MAXi\{Ci\}}{MINi\{Ci\}}$$

presented in [23], [24], the imbalance ratio represents the gap between the data classes which affects the efficiency of the machine learning system. Additionally, sophisticated hackers focus on the development of minority data types to overpass defenses. Therefore, to increase the efficiency of the system, this imbalance rate should be as close as possible to 1 [25].

In other words, imbalance ratio can be defined as the fraction between the number of instances of the majority class (max) and the minority class (min), in our case the majority class is the benign traffic, the minority is the malign one. The calculated imbalance ratio is 1.75, we managed to balance the final DataSet to have a ratio equal to 1 by reducing the number of benign classes.

The two final DataSets are each divided into three sets: Training set, validation set and testing set as,

- 80% of the dataset for training models
- 10% of the dataset for model validation
- 10% of the dataset for model testing

#### 5.2.3. Make and train the various models

The training model is created and is made to fit with different classifiers to check which delivers the best accuracy score, and which classifiers use more features from logistic regression classifier, Naive Bayes classifier, random forest classifier, Gradient boost classifer, decision tree classifier and neural network classifier by using a model.fit () method of sklearn.

#### 5.2.4. Evaluate the various models

We evaluate our models on the Test Set to see the score and the generalisability of them, by comparing predicted and expected outputs. Using sklearn, we get the predictions as,

```
Begin
Prediction= model.predict(inputs_features)
END
```

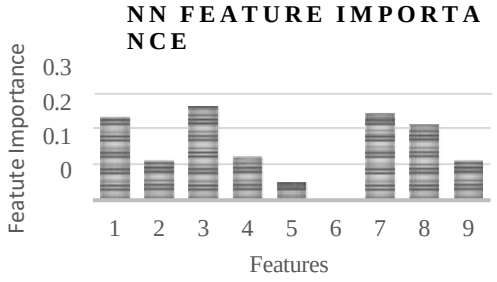
in the scope of this paper, we are interested in understanding how well the produced categories match with those that have been manually classified. In particular, we measured the accuracy, precision and other scores by employing classification report from sklearn.metrics. The coding is as per the following,

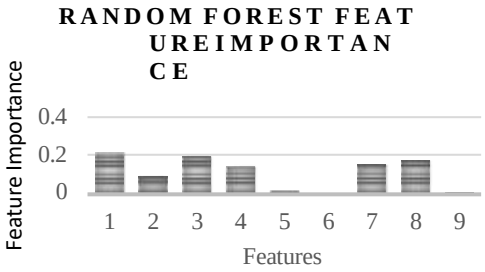
Begin
print(classification_report(expected, predicted);
END

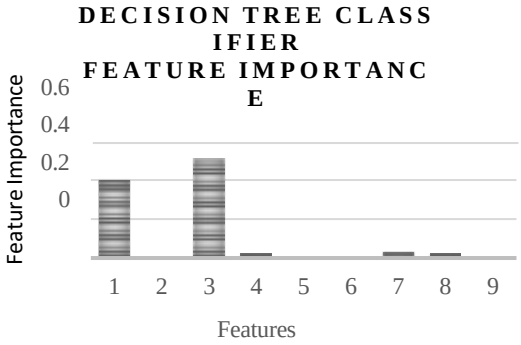
## 6. RESULTS AND ANALYSIS

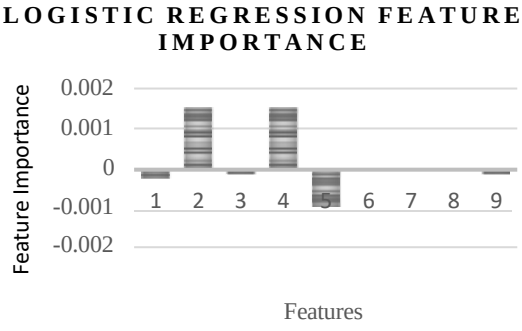
Different statistical metric used for evaluating our models are described as shown in Table 1 and Table 2 for obtained results). We are also interested on features impotence, which play an important role in a predictive modelling project, in which we see if the model takes into account all features or not. Figure 1 and Figure 2 shows the features importance given by the various classifiers over ISOT-CID 2016 and ISOT-CID 2018 respectivelly.

- Accuracy: The Accuracy Score shows how well the model making an exact prediction overall and it is one of the most precise metric scores.
- Recall: Recall could be a measure that tells us how great our model is when all the actual values are positive.
- Precision: It is a recognized actual value from all the expected actual values.
- F1 score: It can be a metric which, by taking its mean value, mixes recall and precision.
- Area Under the Curve (AUC) Score: If the score of AUC is 1, the classifier is ready to distinguish correctly between all the actual values and also the false values.

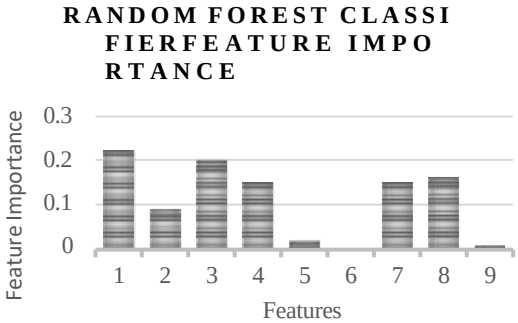






Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 02, February : 2022

Figure 1. Features importances diagrammes for ISOT-CID 2016

An observation of the results of Table 1 shows that most models show good results. The more accurate one is the random forest classifier, reached 100% in all scores, then comes the Gradient boosting classifier and the logistic regression classifier gives the last precision. Graphs of Figure 1 shows also that the random forest classifier is the best in term of use of most of the features during training. Also, the artificial neural classifier is powerful in this side which using many features. In order to check the generalizability of our classifiers, we evaluate them also over ISOT-CID 2018, and the obtained results are as follows:

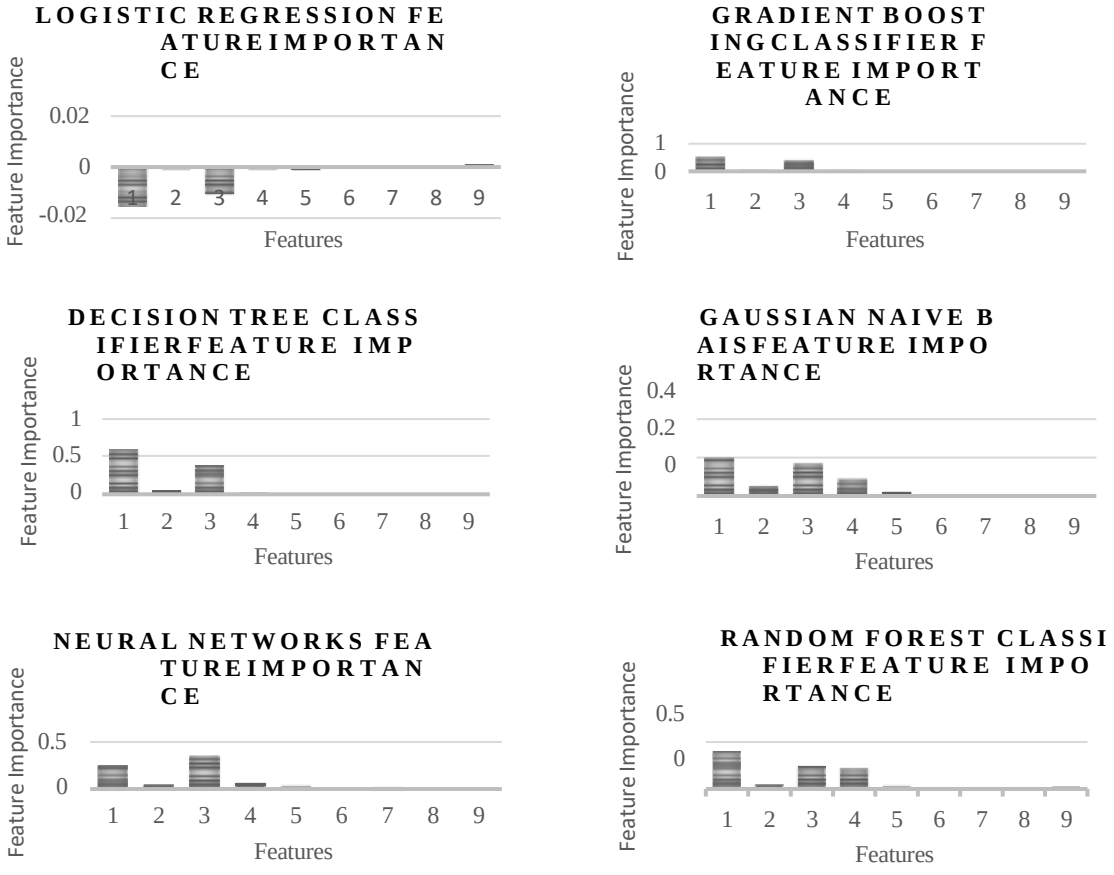



Figure 2. Features importances diagrammes ISOT-CID 2018

Table 1. Accuracy and different metric scores of the various machine learning models over ISOT-CID 2016

| Models | Accuracy | F1 | Precision | Recall | AUC |
|---|---|---|---|---|---|
| Random Forrest Classifier | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Gradient Boosting Classifier | 0.983 | 0.983 | 0.997 | 0.970 | 0.983 |
| Artificial Neural Network Classifier | 0.921 | 0.919 | 0.949 | 0.890 | 0.921 |
| Decision Tree Classifier | 0.911 | 0.902 | 0.997 | 0.824 | 0.911 |
| Naïve Bayes Classifier | 0.768 | 0.764 | 0.777 | 0.751 | 0.768 |
| Logistic Regression | 0.645 | 0.674 | 0.623 | 0.735 | 0.645 |

Table 2 shows also that the most of models gives good results, this time over the DataSet ISOT-CID 2018. The more accurate ones are the artificial neural network classifier and the random forest classifier,

reaching both 100% in all scores, then comes the Gradient boosting classifier and the logistic regression classifier gives the last precision as for 2016 dataset. Figure 2 graphs shows that the artificial neural networks and the random forest classifiers are not only powerful in terms of precision but also in terms of use of most of the features during training. Also, the Naïve Bais classifier is powerful in this side which using many features.

Table 2. Accuracy and different metric scores of the various machine learning models over ISOT-CID 2018

| Models | Accuracy | F1 | Precision | Recall | AUC |
|---|---|---|---|---|---|
| Artificial Neural Network Classifier | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forrest Classifier | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Gradient Boosting Classifier | 0.999 | 0.999 | 0.999 | 1.000 | 0.999 |
| Decision Tree Classifier | 0.996 | 0.996 | 0.992 | 1.000 | 0.996 |
| Naïve Bayes Classifier | 0.959 | 0.958 | 0.986 | 0.931 | 0.959 |
| Logistic Regression | 0.822 | 0.809 | 0.872 | 0.754 | 0.822 |

## 7. CONCLUSION

Cloud computing is considered as a "network of networks" over the internet, this adds complexity to the intrusion detection systems, especially with the particularirty of cloud environment which network access rate are enormous. Different IDS techniques are used to counter malicious attacks in traditional networks but still suffer from lack of perfomance and precision pushing to relinquishing the control of data and applications to service provider. The intelligent intrusion detection models outlined in this paper significantly improve upon the performance of detection methods and achieve perfect sensitivity on the ISOT-CID DataSet. Those models provide excellent precision (minimizing the number of false positives) with two versions of the dataset, which proove their generalisability. As futur work, we plan to train and evaluate our models on other intrusion detection datasets. We plan also to explore the possibilities of integrating the more accurate classifier in a real-time intrusion detection system.

## REFERENCES

[1] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016, doi: 10.1007/s13748-016-0094-0.

[2] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014, doi: 10.1109/TKDE.2012.232.

[3] A. Pharate, H. Bhat, V. Shilimkar, and N. Mhetre, "Classification of Intrusion Detection System," *International Journal of Computer Applications*, vol. 118, no. 7, pp. 23–26, 2015, doi: 10.5120/20758-3163.

[4] D. A. Effendy, K. Kusrini, and S. Sudarmawan, "Classification of intrusion detection system (IDS) based on computer network," *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017*, vol. 2018-January, pp. 90–94, 2018, doi: 10.1109/ICITISEE.2017.8285566.

[5] J. Maestre Vidal, A. L. Sandoval Orozco, and L. J. Garcia Villalba, "Quantitative Criteria for Alert Correlation of Anomalies-based NIDS," *IEEE Latin America Transactions*, vol. 13, no. 10, pp. 3461–3466, 2015, doi: 10.1109/TLA.2015.7387255.

[6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.

[7] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019, doi: 10.1007/s12083-017-0630-0.

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, vol. 2018-January, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[9] A. Aldribi, I. Traore, P. G. Quinan, and O. Nwamuo, "Documentation for the Isot Cloud Intrusion Detection Benchmark Dataset(Isot-Cid)," *University of Victoria*, 2020, [Online]. Available: https://doi.org/10.1016/j.cose.2019.101646.

[10] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Computers and Security*, vol. 88, 2020, doi: 10.1016/j.cose.2019.101646.

[11] R. Rama Devi and M. Abualkibash, "Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper," *International Journal of Computer Science and Information Technology*, vol. 11, no. 03, pp. 65–80, 2019, doi: 10.5121/ijcsit.2019.11306.

[12] J. Mchugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, 2000, doi: 10.1145/382912.382923.

[13] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *2003 Symposium on Applications and the Internet, 2003. Proceedings.*, 2003, vol. 2836, pp. 209–216, doi: 10.1109/SAINT.2003.1183050.

[14] H. Zhang and J. Su, "Naive Bayes for optimal ranking," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 20, no. 2, pp. 79–93, 2008, doi: 10.1080/09528130701476391.

[15] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion detection based on K-Means clustering and Naïve Bayes classification," in *2011 7th International Conference on Information Technology in Asia*, Jul. 2011, vol. 124, pp. 1–6, doi: 10.1109/CITA.2011.5999520.

[16] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.

[17] T. Chen, "Introduction to Boosted Trees," *Data Mining with Decision Trees*, pp. 187–213, 2014, [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/9789812771728_0012.

[18] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, "Fast Gaussian Naïve Bayes for searchlight classification analysis," *NeuroImage*, vol. 163, pp. 471–479, 2017, doi: 10.1016/j.neuroimage.2017.09.001.

[19] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," *Springer*, pp. 157–175, 2012, doi: 10.1007/978-1-4419-9326-7_5.

[20] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of Clinical Epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996, doi: 10.1016/S0895-4356(96)00002-9.

[21] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000, doi: 10.1109/5326.897072.

[22] M. Rocha, P. Cortez, and J. Neves, "Evolution of neural networks for classification and regression," *Neurocomputing*, vol. 70, no. 16–18, pp. 2809–2816, 2007, doi: 10.1016/j.neucom.2006.05.023.

[23] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019,

doi: 10.1186/s40537-019-0192-5.

[24] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.

[25] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets," *2006 IEEE International Conference on Granular Computing*, pp. 732–737, 2006, doi: 10.1109/grc.2006.1635905.