World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:17, No:1, 2023

A Survey in Techniques for Imbalanced Intrusion Detection System Datasets

Najmeh Abedzadeh, Matthew Jacobs

Abstract**—**An intrusion detection system (IDS) is a software application that monitors malicious activities and generates alerts if any are detected. However, most network activities in IDS datasets are normal, and the relatively few numbers of attacks make the available data imbalanced. Consequently, cyber-attacks can hide inside a large number of normal activities, and machine learning algorithms have difficulty learning and classifying the data correctly. In this paper, a comprehensive literature review is conducted on different types of algorithms for both implementing the IDS and methods in correcting the imbalanced IDS dataset. The most famous algorithms are machine learning (ML), deep learning (DL), synthetic minority over-sampling technique (SMOTE), and reinforcement learning (RL). Most of the research use the CSE-CIC-IDS2017, CSE-CIC-IDS2018, and NSL-KDD datasets for evaluating their algorithms.

Keywords**—**IDS, intrusion detection system, imbalanced datasets, sampling algorithms, big data.

I. Introduction

IMBALANCED datasets have been an issue in class distribution in many domains including IDS. This problem is much more important in some domains. For example, detecting a malicious activity as benign can be so costly and we should account for it. Class imbalance happens when one class is insufficiently represented even if it is more of interest. Usually, the minority class is more important, and it is imperative that it is discovered in the dataset. There are many studies related to imbalanced dataset [30], [35], [52]. In this survey, we first discuss about IDS and some of the research around the topic. Then, we explain briefly the most famous IDS datasets in these papers. Next, we present different papers about imbalanced IDS datasets. In the following step, we evaluate several different sampling methods including: undersampling, oversampling, and SMOTE [1]. After that, we have a short section regarding RL, and Generative Adversarial Networks (GANs). Finally, we discuss our findings and conclude the paper.

II. IDS

A. Definition

The evolution of malicious software poses a critical challenge to the design of IDSs. IDS is used to detect a malicious intrusion into a host or network and alerts the user about the potential abnormal behavior in the system. IDS is classified into four different categories: Network-based (NIDS), Host-based (HIDS), Perimeter (PIDS) and VM (VMIDS) based. NIDS is made for monitoring incoming and outgoing network traffic, HIDS monitor and alerts individual host, PIDS detects perimeter intrusions of important system infrastructure, and VMIDS can provide any of the above IDSs or a combination of them as implemented through a virtual machine (VM).

Signature-based detection and anomaly-based detection are two most common specifications of IDS techniques [2]. Signature-based detection works by detecting the known threats, while anomaly-based detection systems compare the normally observed events to identify significant deviations which is useful for detecting previously unknown attacks but suffers from a high false-alarm rate which causes over reactions. An IDS is employed through either scenario approach or behavioral approach. Likewise, signature-based, the scenario approach is based on the comparison between the observed behavior and their corresponding signature for each attack. Like anomaly-based, the behavioral approach is based on the deviation between normal and observed behavior when user has abnormal behavior between his/her usual uses of the system [3].

IDSs have been implemented through several different methods. One of these methods is Software Defined Networking (SDN) which is an architecture that can independently implement dynamic security features [4]. The other technique is Distributed Intrusion Detection System (DIDS) for Wireless Sensor Networks (WSN) which is based on an agent-based, intelligent, and distributed system. DIDS can be set inside the intelligent agents so that they can be located on a network [5]. Even though IDS is implemented through these techniques, ML is more popular in detecting malicious activities. However, research shows that uneven training data can dominate the learning algorithm and therefore malicious cyber-attacks can hide themselves in large imbalanced datasets [5].

B. Implementation

Many of the NIDS are implemented as an open-source software application with different capabilities including Bro (Zeek), Snort, Suricata, Sguil, and many others. Hill et al. utilize the Bro IDS with a simulation model to monitor the system's physical behavior to mitigate unsafe or undesirable system states [6]. Then, the state of the system in the network is compared with the model simulation to observe the inconsistency between them. They do not rely on detecting the attacks directly but instead watching the abnormal behavior of

Najmeh Abedzadeh and Dr. Matthew Jacobs are with EECS Dept., School of Engineering, Catholic University of America, Washington, DC, United States (e-mail: abedzadeh@cua.edu, JacobsMJ@cua.edu).

the system. They can also detect the attacks in the physical portion of the system such as a man in the middle. IDSs are also implemented through ML and DL algorithms in several different research papers. Ziadoon et al. tested the performance of several different ML algorithms in Anomaly Based Intrusion Detection System (AIDS) Datasets and detecting attacks [28]. They reviewed previous studies on AIDS to present suitable algorithms, parameters, and testing criteria. They also measured the true positive and negative rates, accuracy, precision, recall, and F-Score of 31 ML-AIDS models. They mentioned that decision tree (DT), k-nearest neighbor (KNN), and naive Bayes (NB) had the best performance. Finally, they introduced future research in measuring the impact of feature selection in the performance of the ML algorithms.

Satam et al. represented an anomaly-based IDS for the Domain Name System (DNS) protocol (DNS-IDS) for detecting the abnormal behavior of the protocol [8]. DNS is a hierarchical naming system for identifying computers and resources. They first trained the normal behavior of the DNS protocol as a finite state machine to show the normal DNS traffic transition within that state machine. Then, they added some known DNS attacks as abnormal DNS traffic transition and developed an anomaly metric for the DNS protocol for both normal and abnormal behavior using classification algorithms such as bagging. Bagging is an ensemble ML algorithm that prevents overfitting and reduces variance with combining the group of models. It builds N trees in parallel with N randomly generated datasets with replacement. To generalize their method, they evaluated their approach against a wide range of DNS attacks and showed the attack detection rate of 97%.

Ho et al. introduced an IDS based on the Convolutional Neural Network (CNN) for detecting network intrusions by classifying all the packet traffic in the network as benign or malicious classes [9]. They used CICIDS2017 dataset to validate their model in terms of the overall accuracy, attack detection rate, false alarm rate, and training overhead. Finally, they boost the multi-class classification performance of the proposed CNN based IDS and outperformed nine other classifier models such as Hierarchical [10], WISARD [11], Forest PA [12], J48, LIBSVM [13], FURIA [14], Random Forest, MLP, and NB. They represented the highest True Negative Rate (TNR) of 98.984% and the lowest False Alarm Rate (FAR) of 1.015% for benign network traffic along with detecting innovative attacks.

Zoppi et al. demonstrated the capability of unsupervised ML algorithms in implementing an AIDS and detecting cyber-attacks using RELOAD tool [15], [16]. They were able to classify both normal and anomalous behaviors without relying on labeled datasets to detect known attacks, zero-day attacks, and emerging threats.

Mehmood and Rais compared the performance of different supervised ML algorithms including SVM, NB, J.48, and decision table via true positive rate, false positive rate, and precision to detect anomalies in the KDD99 dataset [17]. They demonstrated that none of the algorithms had a high detection rate for each class, but the overall accuracy of J.48 DT was higher among all other algorithms along with low misclassification rate. Also, SVM was the best algorithm for R2L class and NB has the highest FPR among other algorithms.

Dimensionality reduction, clustering, and classification are used in several research papers for anomaly detection [18], [19]. Pervez and Farid applied Support Vector Machine (SVM) on multi-class NSL-KDD Cup99 dataset with combining feature selection and classification [20]. They represented 91% classification accuracy using only three features and 99% classification accuracy using 36 features, while all 41 training features achieved 99% classification accuracy. Shapoorifard and Shamsinejad improved cluster center and nearest neighbor (CANN) intrusion detection methods' classification performance and applied it on NSL-KDD Cup99 dataset using K-Farthest Neighbor (KFN), KNN, and Second Nearest Neighbor (SNN) for classification with measuring the distance between each data sample and each cluster center and calculating the distance between data and its nearest neighbor in the same cluster [21]. Bhattacharya et al. applied a hybrid of PCA and ML algorithms to classify IDS datasets [22]. The KNN, NB, random forest, SVM, and XGBoost algorithms are applied on the reduced dataset taken from an open dataset collected from Kaggle and the original Kaggle dataset before applying PCA and the showed a better performance after using PCA [22].

Parkar and Bilimoria presented a comprehensive paper on different pros and cons of IDS and Intrusion Prevention System considering various techniques such as different types of ML algorithms to make a better choice in selecting the appropriate security model [23]. Gümüşbaş and Yıldırım focus on recent approaches based on DL for IDS [24]. The potential of DL methods for cybersecurity and IDSs and analysis of the benchmark datasets are proposed for providing a road map for readers in IDS. Different types of IDSs specifically network IDS and cloud computing are discussed with their applications and their contribution is described. Zhang et al. proposed a deep hierarchical network on CICIDS2017 dataset and the CTU dataset for integrating the improved LeNet-5 and LSTM neural network structures which trains the hierarchical network at the same time rather than separating them [25].

C. Imbalance Datasets and Metrices

There are many imbalance datasets that are distributed unequally and can create problems in classification. One of the most famous imbalanced datasets is the IDS dataset. Malicious cyber-attacks can hide themselves in large imbalanced datasets which makes it difficult for IDS to detect. In these datasets, one may see the ratio of 8 over 1 for benign vs. malicious activities. Using ML to predict them, we can have a high accuracy, but this cannot be an adequate parameter for validating if the model is working correctly or not as even one malicious activity can destroy the whole system. Some of other parameters that can help us to detect whether our model is a true model are Precision, Recall, F1 score, confusion matrix, the Receiver Operating Characteristic curve (ROC), etc.

Confusion matrix is a matrix of size 2×2 for binary classification with actual values versus predicted values. It has four sections including true positive, true negative, false

negative, and false positive. True Positive (TP) is where prediction and actual both are positive. True Negative (TN) is when prediction and actual both are negative. In False Positive (FP), the predicted is positive while actual is negative and in False Negative (FN), predicted is negative while actual is positive.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$F1 = 2 * \frac{Recall*Precision}{Recall+Precision} \quad (3)$$

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \quad (4)$$

The ROC curve is created by plotting the True Positive Rate (TPR) or recall against the False Positive Rate (FPR).

There are different types of imbalanced datasets including Between-class, Within-class, and Intrinsic and Extrinsic [26]. Within each class, there can be an imbalanced number of data points called between-class imbalance. On the other hand, it can be balanced as a between-class but the range or variation of one or more of the classes is not adequate [26]. IDS is a type of either intrinsic (due to the nature of the dataset) imbalanced dataset or extrinsic (due to time, storage, and other factors). Two of the most famous techniques to overcome any types of imbalanced datasets are sampling methods, and cost-sensitive methods.

Three of the most common sampling methods are undersampling, oversampling, and creating synthetic data. In undersampling, we remove data from the majority classes and in oversampling we generate data for minority classes until all classes have the same number of data points. Undersampling is possible if enough data points are available on the under sampled class. Oversampling is possible if the new synthetic data added are close to the real data. Undersampling resamples the majority class to make the equal to minority. Oversampling resamples the minority class to make the equal to majority. The best method of oversampling is generating new artificial data close in dataspace proximity to existing samples or are 'between' two samples. However, there is a problem with these data points as they are random and there is a risk of overfitting as we may add noise to the dataset instead of actual data. The main problem with oversampling with artificial data is that they do not add any information to dataset. To overcome to this problem, we can use the third method which is creating synthetic data points among previous data points. One of the techniques is using SMOTE. This technique will create new instances between data points in minority class. There is a package in Python called imblearn with a function called over_sampling.SMOTE which create these samples for the dataset [26].

In cost-sensitive learning techniques, either undersampling or oversampling can be done by altering the relative weighting of individual samples. Here are the two most famous ones, upweighting and downweighing:

- Upweighting: It is like oversampling which increases the weight of one of the classes while keeping the weight of the other class.
- Downweighing: It is like oversampling which decreases the weight of one of the classes while keeping the weight of the other class.

III. Datasets

There are five popular datasets used in most of IDS related research [27]. The first two known datasets are CICIDS2017, and the CICIDS2018 with around 2830540 instances [28]. The second famous dataset is the Knowledge Discovery and Data Mining (KDD) Cup 1999 with approximately five million unstructured and raw data with 80% of attack type of data. The third recognized dataset is NSL-KDD dataset [29]. This dataset includes the raw data of KDD Cup 1999 but removes some redundant data [30]. The fourth well-known dataset is the UNSW-NB15 with more than 72.000.000 records [31], and the last known dataset is the AWID dataset with 1,795,575 records of data for training and 575,643 for testing [32]. The following datasets including NSL-KDD, ISCXIDS2012, CICIDS2017, and CICIDS2018 are analyzed using supervised ML algorithms in different papers [33]. Here we discuss a brief description of each one of these datasets.

A. CSE-CIC-IDS2017 and CSE-CIC-IDS2018 Dataset (Canadian Institute for Cybersecurity Intrusion Detection System)

CIC-IDS2018 Dataset includes 10 files from different times of the year all together monitoring traffic for different timestamps in February and March 2018 [28]. Every file has around 79 columns. The last column is the binary label of either "Benign" or "Infiltration", and the other columns are features or predictors for the label column. Some of the files for this dataset are more imbalanced than the others. The most imbalance one was for 02-28-2018 which was binary dataset with 540568 rows of Benign class and 68462 rows of Infiltration class after removing non-sense outliers. Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from the inside are seven of the most famous attack types represented in CSE-CIC-IDS2018 [34]. This dataset is made through 50 machines as attacking infrastructure and the victim organization has five departments with 420 machines and 30 servers. CICFlowMeter-V3 was used to extract the network traffic and system logs of each machine along with 80 features from the traffic such as distributions of packet sizes of a protocol, number of packets per flow, certain patterns in the payload, size of payload, and request time distribution of a protocol. These datasets have been used in several different papers and evaluated using ML algorithms [35], [36], [25].

B. KDD Cup 1999 Dataset

This dataset has been one of the most famous datasets for detecting anomalies since 1999. The data are captured from DARPA'98 IDS evaluation program which is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of

network traffic. It has 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack. The attacks are Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), and Probing Attack. KDD'99 features are divided into the following categories: Basic features, Traffic features ("same host" feature, and "same service" features), and Content features.

C. NSL-KDD Dataset

This dataset is the refined version of KDD-99, more well-organized and cleansed [37], [38], [35]. It has 41 features, including 38 continuous and three categorical variables. They are transformed, normalized, and scaled down to the range of [0 - 1] for the continuous ones and one-hot encoded to dummy variables for the categorical ones. The dataset is grouped into five major categories in the label column: NORMAL, DoS, probing attacks, or PROBE, R2L, U2R. This dataset has been used in several different papers [39].

D. UNSW-NB15 Dataset

This dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra with 2,540,044 records [40]. This is mainly for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. There are nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

E. AWID Dataset

Aegean WiFi Intrusion Dataset (AWID) is newer and larger than NSL-KDD, and is publicly available [32]. This dataset contains 154 features both continuous and categorical with 1.795,574 data samples for training and 575,642 data samples for testing of the datasets, with a class label with four values: normal, flooding, injection, and impersonation with 91% of normal samples and 9% associated with anomalies.

IV. Balancing Algorithms for IDS

A. Balancing Techniques with ML

Data-level approaches and algorithm level approaches are two of types of approaches for solving the problem of imbalanced dataset [41]. At the data-level, the original dataset is balanced using sampling methods algorithm to aid further learning processed. At the algorithm-level, an existing algorithm is modified and strengthened, or a new algorithm is designed on the learning of minority classes to tackle the class imbalance problem.

In an imbalanced dataset, we might expect to achieve an overall high accuracy, but minority classes will suffer from a very low recall score. The false-negative error is not acceptable in some cases such as cancer which may cost a person's life [38]. Deep Neural Network (DNN) has shown improvements over ML models on supervised classification tasks [42], [43]. The network data are complicated and the hierarchy structure of DNN can extract the features of network security very well [44], [45]. However, DL based IDSs cannot detect the network intrusions when we have imbalanced dataset [44].

Subiksha et al. presented a survey about imbalance datasets and how they have been treated in different datasets using ML algorithms for IDS [46]. Of the papers surveyed, 44% were about IDS but not relevant to imbalanced datasets, 22% applied existing methods, 18% proposed a new algorithm, and 16% only mentioned imbalanced dataset. They concluded that most papers have not focused on creating a new algorithm to overcome the problem of imbalance datasets and suggested an opportunity for research work in proposing novel techniques for handling the dataset imbalance.

Lan et al. compared several different classification algorithms such as Random Forest, SVM, XGBoost, LSTM, Mini-VGGNet, and AlexNet to reach a better accuracy [47]. They introduced Difficult Set Sampling Technique (DSSTE) algorithm to reduce the imbalance of training data. First, they divide the imbalance dataset into difficult and easy sets using Edited Nearest Neighbor (ENN) algorithm. The difficult set includes the near-neighbor set and easy set is the far-neighbor set. The data in difficult set are highly similar and it is difficult to differentiate them but the easy set are not similar and easy to distinguish them. Then, K-means algorithm is used to reduce the majority size by replacing some of the majority samples with their K-cluster centroids. Finally, they combine the compressed majority samples with augmented minority samples in difficult samples to make up a new training set. "For oversampling, they zoom in and out the minority in a particular zoom range for different attributes to create new samples in minority and improve classification performance" [47], [48].

Abdul-Hammed et al. use different methods to evaluate the Coburg Intrusion Detection Dataset-001 (CIDDS-001) with 146,500 instances with 99.99% accuracy [49]. These methods are data upsampling, data downsampling, DNN, Random Forest, Voting, Variational Autoencoder and Stacking Machine Learning classifiers. However, their technique is not generalized, and it can have different accuracy for various datasets. The difference between original dataset accuracy and the new dataset accuracy after sampling method is only around 0.03% improvement in average.

To face the problem of imbalance data, Chuang and Wu trained the depth automatic encoder to establish a data generation model for generating balance dataset for NSL-KDD [50]. To generate new data, they apply Deep Variational Autoencoders, which is a neural network that learns how to generate new data like the original input. The generated data conform to the unit Gaussian distribution. The results represent the robustness of the trained model in face of unknown data.

Bedi et al. proposed a new type of IDS based on Siamese Neural Network (Siamese-NN) [51]. The proposed Siam-IDS does not use any balancing techniques for detecting R2L and U2R attacks but the algorithm itself works well for unbalanced datasets [51]. They calculate similarity score between input pairs to identify whether data points belong to the same or different class. Contrastive loss function is also used to maximize the similarity. The proposed algorithm has two identical DNNs comprising of an input layer, five hidden layers and four dropout layers. Comparing their algorithm with other types of algorithms such as DNN and CNN, they were able to

produce better performance. Even though recall had a significant improvement for the two minority classes of the dataset, the authors admitted that the precision was not good enough [51].

Ali et al. proposed a survey of different potential problems with imbalance data and its advancement, including the issues with imbalance classification [52]. They introduce several different approaches in class imbalance classification including data level and algorithm level approaches. The data level approach for handling class imbalance problem is sampling and feature selection. The algorithm level includes improved learning algorithms to handle imbalanced dataset such as z-SVM, or one-class learning algorithms or recognition-based methods that model the classifier on minority class. Cost sensitive learning algorithms are another type where the classifier assigns larger cost to false negatives compared to false positives. Another option is ensemble method which applies several classifiers on training data to decide like boosting and bagging. The last is hybrid approach which employs more than one ML algorithm to achieve better results. Ali et al. mentioned that the best performance evaluation methods are the confusion matrix and its derivations. Raneem et al. proposed a DL approach based on a multi-layer classification method such as SLFN and LSTM [53]. They combined it with an oversampling process for imbalanced IoT devices and offered better performance in terms of Accuracy, G-mean, Precision, and Recall.

B. Undersampling Techniques

There are several different undersampling techniques including ClusterCentroids, RandomUnderSampler, and NearMiss [54]. Undersampling techniques are in two categories prototype generation and prototype selection. Prototype generation such as ClusterCentroids reduces the number of the samples while generates a new set of data from the original dataset. ClusterCentroids generates the new data based on cluster centroid of a K-means algorithm which is synthesized with the centroids of the K-means method instead of the original sample. In prototype selection algorithm, the data are chosen from the original dataset by just reducing the majority not generating new data.

Prototype selection algorithms are separated into two categories called controlled undersampling techniques and cleaning undersampling techniques. The controlled undersampling defines the number of the samples while in cleaning undersampling technique, the algorithm itself cleans the feature space and reduces the noise [54]. RandomUnderSampler, and NearMiss are two types of controlled undersampling techniques. The RandomUnderSampler randomly deletes the rows of the majority and balance the dataset. NearMiss measures the difference between the majority classes and a few specific minority samples [55]. To decrease the number of the rows in majority, it keeps only the majority samples that has smallest average distance to the N nearest/farthest minority samples. Bao et al. updated the weights of negative examples to balance the dataset and applied Boosted NearMiss Undersampling algorithm on the training of SVMs (BNU-SVM) [56].

C. Oversampling Techniques

There are several different oversampling techniques. Random Over Sampling (ROS), SMOTE, Probabilistic Generative Models, RL, and Adversarial Reinforcement Learning are some which we briefly describe them here. For oversampling, the data in minority will be selected to be generated multiple times to make the dataset balanced. For this reason, they keep the values of the columns that play an important key in the prediction fixed and they make a random selection from the remaining data [38].

1) Random Over Sampling

ROS is called “naive resampling” which the “new” samples are duplicated randomly with replacement without considering the nature of data or suing any heuristics [57]. This algorithm randomly selects and duplicates examples from the minority until dataset is balanced. One problem with ROS is overfitting since it makes exact copies from minority class. In this case, a classifier may produce rules that are apparently accurate, but it covers only the replicated examples.

2) Synthetic Minority Over-Sampling Technique

SMOTE [58] is one of the oversampling techniques which generate new samples in between existing data points based on local density and their borders with the other class and generalize the decision region for the minority class [5]. The difference between oversampling and SMOTE is that a typical oversampling technique, the minority class is duplicated from the minority class population. SMOTE algorithm is different as it first finds its k-nearest minority neighbors, then randomly selects j neighbors based on the desired amount of oversampling of these neighbors, and finally randomly generates synthetic samples along the lines joining the minority sample and its j selected neighbors. The problem with SMOTE is that random oversampling makes the decision region for the minority class very specific which leads to overfitting [26]. SMOTE generates new samples on the line segments formed by the endings of its KNN which K is selected based on the amount of required oversampling [38]. Aloul et al. [38] multiply the difference between the feature vector and its nearest neighbor by a random number ranged from 0 to 1 to generate the new samples. In fact, the difference between oversampling and SMOTE is that ROS just increases the size of the training data set through repetition of the original examples without any increase in the variety of training examples while SMOTE not only increases the size of the training data set, but also the variety.

Two of the SMOTE drawbacks are overgeneralization, generalizing the minority area without regard to the majority class, and inflexibility, the number of synthetic samples if fixed [26]. The other issue is that they may introduce the artificial minority class examples too deeply in the majority class space which can be resolved by hybridization: combining SMOTE with undersampling algorithms. SMOTE is improved over several new algorithms. One of them is ENN which removes any example whose class label differs from the class of at least

two of their neighbors. Then they oversample the minority class using SMOTE. The other algorithms are Borderline SMOTE, SVMSMOTE, KMeansSMOTE, Condensed Nearest Neighbor (CNN), Repeated Edited Nearest Neighbor (RENN), and Instance Hardness Threshold (IHT) [26].

SMOTE is not very practical in high dimensional datasets except when feature selection is applied for KNN classifiers [59]. Nitesh et al. combined oversampling and undersampling to improve the performance [60]. They use SMOTE for oversampling through KNN. They multiplied the difference between the feature vector under consideration and its nearest neighbor by a random number between 0 and 1 and added it to the feature vector. However, some minority regions may overlap which increases noise and it is not very practical.

SMOTE has been used in several different research projects to increase the minority or decrease the majority class to avoid biased classification [61]-[63]. Lopez-Martin et al. proposed a new algorithm called Variational Generative Model (VGM) based on a variational autoencoder with 7 variants of SMOTE and ADASYN to the NSL-KDD dataset to generate synthetic data and train several well-known classifiers [39]. Sornxayya et al. proposed a method of three sequential classifiers combination which uses SMOTE to increase the number of minority class in training phase for building classifier model [64]. They trained their models with five different algorithms such as sequential minimal optimization (SMO), J48, IBK, MLP, and NB with 10 cross-validation. The results show an improvement over sensitivity and the accuracy over previous works.

Jeatrakul et al. combined the SMOTE and Complementary Neural Network (CMTNN) on UCI dataset to solve imbalanced data classification and showed improvement on the performance of imbalanced datasets [65]. CMTNN uses a pair of complementary feedforward backpropagation neural networks called Truth Neural Network (Truth NN) and Falsity Neural Network (Falsity NN). Truth NN predicts the degree of the truth memberships with relative probability of true class while Falsity NN predicts the degree of the false memberships which uses the complement outputs of the Truth NN to train the network. Finally, the predicted results of Truth NN and Falsity NN are compared to provide the classification outcomes. For the case of undersampling, Truth NN and Falsity NN are utilized to detect misclassification patterns from a training set and remove them. On the other hand, SMOTE increases several new minority class instances by interpolation.

Yan and Han deal with network traffic imbalance problem with an improved local adaptive composite minority sampling algorithm (LA-SMOTE) based on the DL GRU neural network [66]. Long Short-Term Memory (LSTM) is composed of a cell, an input gate, an output gate and a forget gate. Gated Recurrent Unit (GRU) is a popular variant of LSTM which replaces its three gates with two gates: one is reset gate that determines how to combine the new input information with the previous memory, another one is the update gate that defines how much of the previous information needs to be saved to the current time step. In LA-SMOTE, first KNN is selected from the low frequency samples. Then, high frequency samples are calculated from the selected KNN. Next, based on the size of k, each low frequency sample is assigned to a different region of sample space based on the difference in the number of low-frequency attack samples of the same class in the nearest neighbors.

Abdallah et al. applied SMOTE to balance both binary and multi-class classification datasets and then tested different ML models on top of them and represented XGBoost method with the highest performance [67]. SMOTE is used for balancing the IDS dataset and several different ML algorithms are used to implement multiple IDSs [68]. Then three different Dimensionality Reduction Techniques including PCA, t-SNE, and UMAP are applied to reduce the dataset.

Mariama et al. combined SMOTE and undersampling using Tomek link (TL) and applied two DL models of Long LSTM and CNN for a better IDS on both NSL-KDD and CICIDS2017 datasets and proposed a high accuracy of 99.57% [35]. Tomek link is an undersampling algorithm for cleaning up the overlapping within SMOTE algorithm. It is combined from a pair of examples that belong to different classes from minority and majority that are in each other's nearest neighbor [69]. Jiao et al. proposed SE-DAS (SMOTE and Edited Nearest Neighbors with Dual Attention SRU, SEDAS), which uses the SE algorithm for balancing the UNSW-NB15 dataset [41]. To make the model more stable, a timing attention mechanism is used for selecting the historical information at significant time points in the Simple Recurrent Units (SRU) network. Karatas et al. applied SMOTE to balance CSE-CIC-IDS2018 dataset and then six different machine-learning-based IDSs are proposed to increase the detection rate for rarely encountered intrusions [36].

SMOTE is used on UCI depository to overcome imbalanced dataset and used single-layered complex valued neural network (CVNN) to classify them and showed a better sensitivity and accuracy [70]. Yan et al. proposed Region Adaptive Synthetic Minority Oversampling Technique (RA-SMOTE) and tested the effectiveness of the algorithm using different types of classifiers, including SVMs, BP neural network (BPNN), and random forests (RF) [71]. Tallo et al. proposed the SMOTE-Simple Genetic Algorithm (SMOTE-SGA) method to overcome the problem of overgeneralization in SMOTE by determining the sampling rate of each instance in unequal amounts of synthetic instances and compared them using G-means and F-Measure [72].

Kurniawan et al. applied C5.0 algorithm from Data Mining to forecast rainfall in Bandung Regency and used SMOTE to overcome the imbalance dataset [73]. Using k-fold cross-validation for validating the data, they showed a high accuracy of 99% after using SMOTE. Jimoh et al. proposed J48 DT ML algorithm with application of SMOTE technique on CICIDS2017 and showed an accuracy of 99.85% [74]. Lu et al. used RF Classifier to integrate the SMOTE with the ENN Rule and represented a higher precision, recall and F1-value compared with previous works [75].

3) Data Augmentation Methods

a) Probabilistic Generative Models

The probabilistic generative models such as Markov chain Monte Carlo (MCMC) have been widely used in sampling the parameters of observed data and approximating the distribution of the data [76]. MCMC is based on Metropolis-Hastings (MH) algorithm [77] which was improved in Gibbs sampling [78], [79] and finally a new version of it was introduced as expectation maximization (EM) [80]. The difference between them was that MH generates data from a proposal distribution, Gibbs uses a full conditional distribution, and EM estimates the distribution between an expectation step and a maximization step.

Abedzadeh et al. applied MCMC, GANs algorithm, and oversampling to balance the CSE-CIC-IDS2018 dataset. Comparing different ML algorithms with these datasets, they represented that Logistic Regression with original dataset was the fastest with accuracy of 0.88 and recall of 0.99 [81].

Zhang et al. integrated adversarial and statistical learning to generate synthesized intrusion data [82]. The data were first synthesized using MCMC algorithms and then augmented by deep generative neural networks through adversarial learning. Poisson-Gamma joint probabilistic model (PGM) generates synthesized intrusion data with some samples from network intrusion data which can be combined with real data as an input to be trained into Deep Generative Neural Networks (DGNNs) and produce the augmented network intrusion data [82]. The results show an improved accuracy in finding the most intrusion types except snmpgetattack attack and snmpguess attack of R2L [82].

b) Reinforcement Learning (RL)

RL is a type of ML that has four components, the agent, the action, the environment, and the reward state [38]. It can develop behaviors through trial-and-error simulations with a dynamic environment. The process of RL has an agent that gets rewarded based on its actions with a recursive learning process [37]. The reward function is manually controlled and is not generated by the environment itself, and the real-time generation of sequences of actions, states and rewards could be assimilated to those registered in a dataset. For example, Google has RL agents that learn to solve the problems by taking different actions. Agents have no prior knowledge at the beginning but learn by trying different moves randomly at the first and learn by getting reward from taking actions that is closer to the best results.

Comparing supervised and unsupervised learning with RL, RL works better when data are large enough, it is in a real time context, and without consistent adjustment of labels that agent can self-learn on its own without any supervising activities during the learning process [38]. Instead of giving rules, or calculating the similarities and differences between data points, RL is based on rewards and punishments as instructions for positive and negative behavior [38]. A software agent constantly interacts with the environment and learns from the acquired rewards and punishments, and it is sequential considering long-term accumulative reward while supervised and unsupervised learning considers instant reward. The environment agent learns the prediction performance of classifier agent and for the next classifier agent, it selects the best categories of data for training [38]. Then, the dataset is balanced as the classifier agent is always forced to train on the most difficult samples at the moment.

Arturo Servin detected Flooding-Base Distributed Denial of Service Attacks using RL and tile coding [83]. They propose a simulated network environment, where they control the injection of anomalies and provide reward based on the correct detection of the anomalies. The algorithm for RL is Q-learning algorithm which works based on a look-up table. To avoid unlimited increase of table size, they discretize the states. There are two researches in RL based on a multi-agent architecture that perform intrusion detection. They also use look up table to discretize the states [83], [84].

Deep Reinforcement Learning (DRL) had attracted significant interest due to its ability to learn complex behaviors in high dimensional data space [7], [85]-[88]. Huang et al. designed a time series anomaly detector using DRL [7]. The model made no assumption about the underlying mechanism of anomaly patterns and adapted to dynamic environments. It also made RL simpler by removing the threshold settings.

Caminero et al. [89] proposed a framework called AE-RL which implements a classifier based on the theory of RL where the behavior of the environment is adjusted in parallel with the learning process. First, they produce random data from training and generate rewards based on the performance of the classifier. Then, they adjust this initial behavior with an adversarial objective and increase the difficulty of the prediction made by the classifier.

Elderman et al. used Monte Carlo and Q-learning to simulate a network environment as a cybersecurity zero-sum game. In the proposed method, two adversarial agents acting as attacker and defender try to win by providing more effective learning models [90]. Zhu et al. applied an ad-hoc RL algorithm based on numerical simulations to introduce an adversarial environment based on RL. The simulated environment has a defender and attacker which dynamically adjust each other's behavior [91].

c) Adversarial Reinforcement Learning

Generative Adversarial Networks (GANs)

GANs [92] have been used extensively in computer vision which have two types of structured DNNs called discriminative and generative [93], [94]. In this structure, the generative DNN tries to generate new data interpolated between training data and discriminative learns to distinguish between random and real data and make the generator to generate data closer to real ones using gradient feedback of generator and finally produce a powerful data generator. As both generative and discriminative are constructed from DNN, they have training difficulties and one of them can get worse while the other one gets better as they have different convergence speed [95], [96].

Peng et al. proposed Sample Equalization for Intrusion Detection System (SE-IDS) which uses GANs to balance the

dataset and then optimized the parameters of LightGBM by applying particle swarm optimization (PSO) on the industrial network dataset [97]. Fadi et al. used Adversarial Autoencoders (AAE) which are a probabilistic autoencoder based on GANs [98]. Then they trained the AAE with the KNN algorithm on the NSL-KDD intrusion data set and proposed a high accuracy of 99.991% [99].

V. Discussion

In this research, we evaluated not only IDS, but different algorithms to resolve imbalanced IDS datasets. Different papers introduced various ML algorithms for detecting malicious activities in imbalanced datasets, but recent papers showed that the combination of DRL, ML, and sampling methods is more effective than other algorithms. We learned about various sampling methods including undersampling, oversampling, and SMOTE. We also discussed other algorithms for generating data for imbalanced datasets such as RL, GANs. There is much research on using ML algorithms for detecting malicious activities in IDS datasets and some other papers about solving the problem of imbalanced IDS datasets. RL and GANs are new techniques which have been recently worked to overcome these dilemmas. As detecting even one attack as benign is so costly for every system and most of the IDS datasets are imbalanced, there are much new research that can be done to face this issue. Regarding the new papers in RL, it is believed that the combination of RL, and sampling methods can be a good area of research about imbalanced IDS datasets. We also learned about most famous datasets for evaluating imbalanced datasets including CSE-CIC-IDS2017 dataset, CSE-CIC-IDS2018 dataset, AWID, KDD Cup 1999, and NSL-KDD dataset.

VI. Conclusion

In this paper, we completed a comprehensive survey around different techniques in IDS, various algorithms to overcome imbalanced IDSs, and a variety of IDS datasets. The KNN, CNN, SVM, LSTM, RF, SVM, XGBoost, DT, and Neural Network were the most famous ML algorithms in implementing the IDS. Based on the type of the malicious activities and the type of the dataset, one of these above-mentioned algorithms work better. To overcome the imbalanced datasets, the following algorithms were applied mostly including ML, DL, SMOTE, NearMiss undersampling, Probabilistic Generative Models, and RL.

References

[1] Vaibhav Jayaswal, "Dealing with Imbalanced dataset", https://towardsdatascience.com/dealing-with-imbalanced-dataset-642a5f6ee297, Oct 18, 202

[2] K. Scarfone, "Guide to intrusion detection and prevention systems (idps)," Comput. Secur. Res. Center, 2012.

[3] L. Dali et al., "A survey of intrusion detection system," 2015 2nd World Symposium on Web Applications and Networking (WSWAN), 2015, pp. 1-6, doi: 10.1109/WSWAN.2015.7210351.

[4] S. Seeber and G.D. Rodosek, "Towards an adaptive and effective IDS using OpenFlow", *IFIP International Conference on Autonomous Infrastructure Management and Security*, pp. 134-139, 2015, June.

[5] A.K. Sharma, S.K. Saroj and P. Kumar, "Distributed intrusion detection system for wireless sensor networks", *IOSR Journal of Computer Engineering*, vol. 14, no. 4, pp. 61-70, 2013.

[6] Zachary Hill, John Hale, Mauricio Papa, and Peter J. Hawrylak, "Using Bro with a Simulation Model to Detect Cyber-Physical Attacks in a Nuclear Reactor" 2019 2nd International Conference on Data Intelligence and Security (ICDIS), 2019

[7] C. Huang, Y. Wu, Y. Zuo, K. Pei, and G. Min, "Towards experienced anomaly detector through reinforcement learning," in Proc. Thirty-Second AAAI Conf. Artif. Intell. (AAAI-18), (Hilton New Orleans Riverside, USA), 2018.

[8] Pratik Satam, Hamid Alipour, Youssif Al-Nashif, and Salim Hariri, "DNS-IDS: Securing DNS in the Cloud Era" 2015 International Conference on Cloud and Autonomic Computing 2015.

[9] Samson Ho, Saleh Al Jufout, Khalil Dajani, and Mohammad Mozumdar, "A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks using Convolutional Neural Network ". IEEE Open Journal of the Computer Society, 2021.

[10] Ahmim, Ahmed, et al. "A novel hierarchical intrusion detection system based on decision tree and rules-based models," In Proceedings of IEEE 15th International Conference on Distributed Computing in Sensor Systems, pp. 228-233, 2019.

[11] M. D. Gregorio and M. Giordano, "An experimental evaluation of weightless neural networks for multi-class classification," Applied Soft Computing, vol. 72, pp. 338–354, 2018.

[12] M. N. Adnan and M. Z. Islam, "Forest PA: Constructing a decision forest by penalizing attributes used in previous trees," Expert Systems with Applications, vol. 89, pp. 389–403, 2017.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 1–27, 2011.

[14] Zhang Xueqin, Chen Jiahao, Zhou Yue, Han, Liangxiu, Lin Jiajun, "A Multiple-layer Representation Learning Model for Network-Based Attack Detection," IEEE Access, pp. 1-1. 2019.

[15] Tommaso Zoppi, Andrea Ceccarelli, Andrea Bondavalli, "Into the Unknown: Unsupervised Machine Learning Algorithms for Anomaly-Based Intrusion Detection ". 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S), 2020.

[16] Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2019, October). "Evaluation of Anomaly Detection algorithms made easy with RELOAD" In Proceedings of the 30th Int. Symposium on Software Reliability Engineering (ISSRE), pp 446-455, IEEE

[17] Tahir Mehmood and Helmi B Md Rais, "Machine Learning Algorithms In Context Of Intrusion Detection" 2016 3rd International Conference On Computer And Information Sciences (ICCOINS), 2016.

[18] D. A. Cieslak, N. V. Chawla, and A. Striegel, ''Combating imbalance in network intrusion datasets,'' in Proc. IEEE Int. Conf. Granular Comput., May 2006, pp. 732–737.

[19] M. Zamani and M. Movahedi, ''Machine learning techniques for intrusion detection,'' 2013, arXiv:1312.2177. (Online). Available: http://arxiv.org/abs/1312.2177

[20] M. S. Pervez and D. M. Farid, ''Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs,'' in Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA), Dec. 2014, pp. 1–6.

[21] H. Shapoorifard and P. Shamsinejad, ''Intrusion detection using a novel hybrid method incorporating an improved KNN,'' Int. J. Comput. Appl., vol. 173, no. 1, pp. 5–9, Sep. 2017.

[22] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, ''A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU,'' Electronics, vol. 9, no. 2, p. 219, Jan. 2020.

[23] P. Parkar and A. Bilimoria, "A Survey on Cyber Security IDS using ML Methods," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 352-360, doi: 10.1109/ICICCS51141.2021.9432210.

[24] D. Gümüşbaş, T. Yıldırım, A. Genovese and F. Scotti, "A Comprehensive Survey of Databases and Deep Learning Methods for Cybersecurity and Intrusion Detection Systems," in IEEE Systems Journal, vol. 15, no. 2, pp. 1717-1731, June 2021, doi: 10.1109/JSYST.2020.2992966.

[25] Y. Zhang, X. Chen, L. Jin, X. Wang and D. Guo, "Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data," in IEEE Access, vol. 7, pp. 37004-37016, 2019, doi: 10.1109/ACCESS.2019.2905041.

[26] Matthew Stewart, "Guide to Classification on Imbalanced Datasets", https://resources.experfy.com/ai-ml/imbalanced-datasets-guide-

classification/, December 1, 2020.

[27] C. Vij and H. Saini, "Intrusion Detection Systems: Conceptual Study and Review," *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2021, pp. 694-700, doi: 10.1109/ISPCC53510.2021.9609481.

[28] Ziadoon Kamil Maseer, Robiah Yusof, Nazrulazhar Bahaman, Salama A. Mostafa, and Cik Feresa Mohd Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," IEEE, VOLUME 9, February 3, 2021.

[29] Subiksha Srinivasa Gopalan1, Dharshini Ravikumar1, Dino Linekar, Ali Raza1, Maheen Hasib, "Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset," 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), IEEE, 2021.

[30] Lan Liu, Pengcheng Wang, Jun Lin, and Langzhou Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning," vol. 9, December 2021

[31] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.

[32] C. Kolias et al. "Intrusion Detection in 802.11 Networks", IEEE communication surveys & tutorials, vol. 18, no. 1, first quarter 2016

[33] Laurens D'hooge **(**UGent**)**, Tim Wauters **(**UGent**)**, Bruno Volckaert **(**UGent**)** and Filip De Turck **(**UGent**)** "Classification Hardness for Supervised Learners on 20 Years of Intrusion Detection Data." IEEE ACCESS, vol. 7, 2019, pp. 167455–69.

[34] A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018) was accessed on 01/01/2022 from https://registry.opendata.aws/cse-cic-ids2018.

[35] Mbow, M., Koide, H., & Sakurai, K. (2021). An Intrusion Detection System for Imbalanced Dataset Based on Deep Learning. In *Proceedings - 2021 9th International Symposium on Computing and Networking, CANDAR 2021* (pp. 38-47). (Proceedings - 2021 9th International Symposium on Computing and Networking, CANDAR 2021). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CANDAR53791.2021.00013

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

[37] Xiangyu Ma and Wei Shi, "AESMOTE: Adversarial Reinforcement Learning with SMOTE for Anomaly Detection", IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, April-June 2021

[38] Fadi Aloul, Imran Zualkernan, Nada Abdalgawad, Lana Hussain, Dara Sakhnini, "Network Intrusion Detection on the IoT Edge Using Adversarial Autoencoders," 2021 International Conference on Information Technology (ICIT).

[39] M Lopez-Martin, B. Carro and A Sanchez-Esguevillas, "Variational data generative model for intrusion detection". Knowledge and Information Systems (2018). https://doi.org/10.1007/s10115-018-1306-7

[40] X. Jiao and J. Li, "An Effective Intrusion Detection Model for Class-imbalanced Learning Based on SMOTE and Attention Mechanism," 2021 18th International Conference on Privacy, Security and Trust (PST), 2021, pp. 1-6, doi: 10.1109/PST52912.2021.9647756.

[41] J. L. Leevy, T. M. Khoshgoftaar, B. R. A., and S. N., "A survey on addressing high-class imbalance in big data," J. Big Data, vol. 5, no. 42, 2018.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Int. Conf. Learn. Represent., 2015.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2016, pp. 770–778.

[44] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Top. Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018.

[45] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in Proc. Int. Conf. Wirel. Netw. Mob. Commun., 2016, pp. 258–263.

[46] "NSL-KDD dataset," https://www.unb.ca/cic/datasets/nsl.html, Canadian Institute of Cybersecurity

[47] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," *Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

[48] H. Hota and A. Shrivas, "Data Mining Approach for Developing Various Models Based on Types of Attack and Feature Selection as Intrusion Detection Systems (IDS)," Intelligent Computing, Networking, and Informatics, 845–851, 2014.

[49] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Abu Mallouh, ``Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE sensors Lett., vol. 3, no. 1, Art. no. 7101404, Jan. 2019.

[50] P.-J. Chuang and D.-Y.Wu, ``Applying deep learning to balancing network intrusion detection datasets," in Proc. IEEE 11th Int. Conf. Adv. Infocomm Technol. (ICAIT), pp. 213_217, Oct. 2019.

[51] P. Bedi, N. Gupta, and V. Jindal, ``Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network," Procedia Comput. Sci., vol. 171, pp. 780_789, 2020.

[52] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," Int. J. Adv. Soft Comput. Its Appl., vol. 7, no. 3, pp. 176–204, 2015.

[53] R. Qaddoura, A. M. Al-Zoubi, I. Almomani and H. Faris, "Predicting Different Types of Imbalanced Intrusion Activities Based on a Multi-Stage Deep Learning Approach," 2021 International Conference on Information Technology (ICIT), 2021, pp. 858-863, doi: 10.1109/ICIT52682.2021.9491634.

[54] Hasan Ersan YAĞCI, "https://hersanyagci.medium.com/under-sampling-methods-for-imbalanced-data-clustercentroids-randomundersampler-nearmiss-eae0eadcc145," Jul 15, 2021

[55] S. J. Yen and Y. S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," Intell. Control Automat., vol. 344, pp. 731–740, 2006.

[56] L. Bao, C. Juan, J. Li, and Y. Zhang, "Boosted near-miss under-sampling on svm ensembles for concept detection in large-scale imbalanced datasets," Neurocomputing, vol. 172, pp. 198–206, 2016.

[57] Y. Kamei, A. Monden, S. Matsumoto, T. Kakimoto, and K. Matsumoto, "The effects of over and under sampling on fault-prone module detection," in Proc. First Int. Symp. Empirical Softw. Eng. Meas., (Madrid, Spain), 2007.

[58] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''SMOTE: Synthetic minority over-sampling technique,'' J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002

[59] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," Blagus Lusa BMC Bioinf., vol. 14, p. 106, 2013.

[60] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research 16 (2002) 321–357

[61] Sornxayya Phetlasy, Satoshi Ohzahata, Celimuge Wu, and Toshihito Kato, "Applying SMOTE for a Sequential Classifiers Combination Method to Improve the Performance of Intrusion Detection System," IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, 2019

[62] N. Qazi, and K. Raza, "Effect of feature selection, synthetic minority over-sampling (SMOTE) and under-sampling on class imbalance classification," 14th International conference on modelling and simulation, pp. 145-150, 2012.

[63] A. Tesfahun, and D. L. Bhaskari, "Intrusion detection using random forests classifier with SMOTE and feature reduction," International conference on cloud & ubiquitous computing & emerging technologies, pp. 127-132, 2013

[64] Y. Sun, and F. Liu, "SMOTE-NCL: A re-sampling method with filter for network intrusion detection," 2nd IEEE International conference on computer and communications, pp. 1157-1161, 2016.

[65] P. Jeatrakul, K. W. Wong, and C. C. Fung, ''Classification of imbalanced data by combining the complementary neural network and smote algorithm,'' in Proc. Int. Conf. Neural Inf. Process. Springer, 2010, pp. 152–159.

[66] B. Yan and G. Han, ''LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network,'' Secur. Commun. Netw., vol. 2018, pp. 1–13, Aug. 2018.

[67] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021, doi: 10.1109/ACCESS.2021.3120626.

[68] T N Varunram, Shivaprasad M B, Aishwarya K H, Anush Balraj, Savish S V, and Ullas S, " Analysis of Different Dimensionality Reduction Techniques and Machine Learning Algorithms for an Intrusion Detection

System," 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA).
[69] G. Batista, B. Bazzan, M. Monard, "Balancing Training Data for Automated Annotation of Keywords: A Case Study," In WOB, 10-18, 2003
[70] K. Matsuda and K. Murase, "Single-Layered Complex-Valued Neural Network with SMOTE for Imbalanced Data Classification," 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS), 2016, pp. 349-354, doi: 10.1109/SCIS-ISIS.2016.0079.
[71] B. Yan, G. Han, M. Sun and S. Ye, "A novel region adaptive SMOTE algorithm for intrusion detection on imbalanced problem," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017, pp. 1281-1286, doi: 10.1109/CompComm.2017.8322749.
[72] T. E. Tallo and A. Musdholifah, "The Implementation of Genetic Algorithm in Smote (Synthetic Minority Oversampling Technique) for Handling Imbalanced Dataset Problem," 2018 4th International Conference on Science and Technology (ICST), 2018, pp. 1-4, doi: 10.1109/ICSTC.2018.8528591.
[73] Erwin Kurniawan, Fhira Nhita, Annisa Aditsania, and Deni Saepudin, "C5.0 Algorithm and Synthetic Minority Oversampling Technique (SMOTE) for Rainfall Forecasting in Bandung Regency," 2019 7th International Conference on Information and Communication Technology (ICoICT).
[74] Ilyas Adeleke Jimoh, Idris Ismaila, and Morufu Olalere, "Enhanced Decision Tree - J48 With SMOTE Machine Learning Algorithm for Effective Botnet Detection in Imbalance Dataset," 15th International Conference on Electronics Computer and Computation (ICECCO 2019).
[75] T. Lu, Y. Huang, W. Zhao and J. Zhang, "The Metering Automation System based Intrusion Detection Using Random Forest Classifier with SMOTE+ENN," 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), 2019, pp. 370-374, doi: 10.1109/ICCSNT47585.2019.8962430.
[76] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," Mach. Learn., vol. 50, no. 1-2, pp. 5–43, 2003.
[77] N. Metropolis and S. Ulam, "The Monte Carlo method," J. Am. Stat. Assoc., vol. 44, no. 247, pp. 335–341, 1949.
[78] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 5-6, pp. 721–741, 1984.
[79] K. P. Murphy, Machine learning: A probabilistic perspective, The MIT Press., 2012.
[80] G. C. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," J. Am. Stat. Assoc., vol. 85, no. 411, pp. 699–704, 1990.
[81] N. Abedzadeh, M. Jacobs, "Using Markov Chain Monte Carlo Algorithm for Sampling Imbalance IDS Datasets", The 31st International Conference on Computer Communications and Networks (ICCCN 2022), July 25 - July 28, 2022, Submitted for review.
[82] He Zhang, Xingtui Yu, Han Xiao, Peng Ren, Chunbo Luo, and Geyong Min, "Deep Adversarial Learning in Intrusion Detection: A Data Augmentation Enhanced Framework" draft paper, January 2019. Preprint available in link "https://arxiv.org/pdf/1901.07949.pdf" downloaded January 2022.
[83] A. Servin, "Multi-Agent Reinforcement Learning for Intrusion Detection". PhD thesis, University of York. 2009.
[84] K. Malialis, "Distributed Reinforcement Learning for Network Intrusion Response", PhD thesis, University of York. 2014.
[85] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," IEEE Internet Things J., 2020.
[86] M. A. Wiering et al., "Reinforcement learning algorithms for solving classification problems," IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), Paris, 2011, pp. 91-96.
[87] M. G. Lagoudakis and R. Parr, "Reinforcement Learning as Classification: Leveraging Modern Classifiers" In Proceedings of the 20th International Conference on Machine Learning (ICML-03), 424–431, 2003, Washington, DC, USA.
[88] K. H. Quah, C. Quek and G. Leedham, "Pattern classification using fuzzy adaptive learning control network and reinforcement learning," Proceedings of the 9th International Conference on Neural Information Processing. ICONIP '02., Singapore, 2002, pp. 1439-1443 vol.3. 2002.
[89] G. Caminero and B. Lopez-Martin, M. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," Comput. Netw., vol. 159, pp. 96–109, 2019.
[90] R. Elderman et al., "Adversarial Reinforcement Learning in a Cyber Security Simulation" International Conference on Agents and Artificial Intelligence (ICAART 2017).
[91] M. Zhu, Z. Hu and P. Liu, "Reinforcement Learning Algorithms for Adaptive Cyber Defense against Heartbleed" Proceedings of the First ACM Workshop on Moving Target Defense, Pages 51-58, 2014.
[92] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[93] H. Zhang, C. Luo, X. Yu, and P. Ren, "Mcmc based generative adversarial networks for handwritten numeral augmentation," in Proc. Int. Conf. Commun. Signal Process. Syst., 2017, pp. 2702–2710.
[94] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 82–90
[95] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.
[96] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in Int. Conf. Learn. Represent., 2017.
[97] Peng Shi, Xuebing Chen, Xiangying Kong, and Xianghui Cao, "SE-IDS: A Sample Equalization Method for Intrusion Detection in Industrial Control System," 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2021.
[98] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," ArXiv151105644 Cs, May 2016. Available at: http://arxiv.org/abs/1511.05644.
[99] Fadi Aloul, Imran Zualkernan, Nada Abdalgawad, Lana Hussain, Dara Sakhnini, "Network Intrusion Detection on the IoT Edge Using Adversarial Autoencoders," 2021 International Conference on Information Technology (ICIT)