Data Resampling Technologies Applied to DDoS Attack Detection of IoT Devices with Machine Learning Application

Yang Ye[1], Jiaxin Zhao[2], Yifan Zhan[3]

[1]Shandong University in Weihai, Weihai, Shandong 264200, China.

[2]Temple University, Philadelphia, Pennsylvania 19122, America.

[3]Xiamen University, Xiamen, Fujian 361000, China.

Abstract

With increasing Internet of Things device connected to the Internet, security problems have raised awareness. Distributed denial of service (DDoS) is a kind of common and dangerous attack towards particular Internet infrastructure, which arouses the necessity to develop powerful technologies to detect such attack. Former researches have discussed various detection models. However, due to lack of big data in the real world, some researches built their models using simulative data, which has imbalance between different classes. This project applies data resampling technologies to solve the problem of imbalance and compares the detection ability of some widely used machine learning applications using the balanced data. The results indicate that using balanced data help train models better and machine learning algorithms can detect DDoS attack.

Keywords

Internet of Things, DDoS, data resampling, machine learning.

1. INTRODUCTION

Internet of Things (IoT) is a network-connected device's technology that has billions of connected devices collecting and sharing data. With the capability to collect, quantify and understand the surrounding environments, an increasing number of IoT devices are replacing traditional non-networked products. It is conjectured that the number of IoT devices will reach 75 billion by 2030[1]. Yet, many devices are faced with security problems and their owners are unaware of the vulnerabilities of these IoT devices.

Because of the insecurity and rapidly expanding the diversity of IoT devices, IoT malware such as Mirai has used insecure IoT devices to conduct botnet attacks. For instance, Distributed Denial-of-Service (DDoS) attacks are a rising threat of IoT devices, which is a group attack initiated simultaneously by hundreds or even thousands of hosts with attack processes installed after the invasion.

This rising threat stimulates the development of detecting DDoS attack traffic techniques from IoT botnets. A recent survey has retrospected research on IoT security problems and found machine learning and deep learning promising in intrusion detection[2]. Doshi, Apthorpe, & Feamster have designed an experiment that simulates the DoS attack on IoT devices of 2018, and they applied machine learning algorithms to anomaly detection to the simulated traffic data to identify whether the devices are attacked [3]. They have achieved satisfactory results and the accuracy of their models reaches approximately 0.99. However, there are two issues in their research. Firstly, the number of attack traffic is almost 12 times bigger than the number of normal traffic, which causes the imbalance of data and might influence the accuracy of the algorithm. Secondly, instead of using data from the real-world, the data is generated from some equipment in order to simulate a DDoS attack.

Based on the two issues, this project conduct some data pre-processing and perform the same experiments, including data collection, feature extraction, and binary classification for IoT traffic DDoS detection.

2. BACKGROUND AND RELATED WORK

In this section, a brief background on network anomaly detection and data pre-processing is presented.

2.1. Network Anomaly Detection

Anomaly detection attempts to classify deviation from the expected behavior. Our work aims to develop anomaly detection techniques that can be used to distinguish DDoS attack traffic from regular traffic. V. Chandola et al. suggest that simple threshold-based techniques are not suitable for anomalous traffic classification because they are prone to classifying normal traffic as attack traffic[4]. More sophisticated algorithms, such as machine learning(ML) and deep learning(DL), have advanced considerably in anomaly detection. In virtue of big data, such approaches can help raise accuracy and minimize false positives.

A survey has retrospected potential ML methods for securing IoT systems [2]. Since anomaly detection is part of the security problems, the literature can provide some suitable ML algorithms for us, such as Random Forest, Support Vector Machines, and Nearest Neighbor Classifiers.

Our hypothesis that IoT traffic is different from other types of networks is the same as the original research which states that the hypothesis has been proved by literature.

2.2. Imbalanced Data

The majority of machine learning algorithms hypothesize that the data set is well-balanced and thus each misclassification contributes equal error. However, well-balanced data in the real world is hard to find and imbalanced data might cause a severe bias problem. To deal with this contradiction, particular methods have been used to improve the quality of their data set.

For instance, Chawla et al. use Synthetic Minority Over-sampling Technique(SMOTE) and compare the effect with other approaches[5]. He et al. Introduce Adaptive Synthetic Sampling Approach(ADASYN) for imbalanced learning[6]. Jaedong Lee and Jee-Hyong Lee utilize the K-means Clustering algorithm and then balance each cluster's data by sampling techniques[7]. By far, over-sampling and under-sampling with machine learning applications are the most used methods. Few works have tried to use deep learning application to help balance the data set.

Based on the observation, we test the common methods and new data-resampling method with deep learning applications, compare their performance and then give comments on each method.

3. MATERIAL AND METHOD

3.1. Data Manipulation Methods

As mentioned above, due to the predominant proportion of the class of attack (93%), the data is extremely imbalanced. Imagine if we have a classifier that classifies each traffic flow to be attacked, it still yields 93 percent accuracy on the test set regardless of the input data. The model would be seriously biased towards the majority class and reduce the performance and reliability of the classification models. To solve this issue, we decided to apply the resampling methods to the data.

Under-sampling and over-sampling are the most common remedies of the imbalanced data issue.

1. Under-sampling techniques balance the dataset by reducing the size of the majority class. In this paper, we will use Random under-sampling, Cluster centroids, Tomek links, Neighbourhood cleaning rule, and Near-miss methods to do under-sampling. The random under-sampling is the simplest under-sampling method that is to under-sample the majority class randomly and uniformly. The cluster centroid method applies the K-means clustering algorithm to replace the majority clusters with the centroid of themselves. A pair of Tomek links are two observations that are each other's nearest neighbors in different classes. In the Tomek links method, these pairs could be removed to reduce the size. The Neighborhood cleaning rule finds three-nearest neighbors of each observation and removes all misclassified examples, then it removes the positively classified observations belonging to the majority class. In this paper, the NearMiss method is Near Miss-1, which selects samples from the majority class for which the average distance of the K nearest samples of the minority class is the smallest.

2. The over-sampling technique replicates the minority class. First, we are going to perform three over-sampling techniques: Random over-sampling, Synthetic Minority Oversampling Technique (SMOTE), and Adaptive Synthetic Sampling(ADASYN). Random over-sampling is the baseline. The SMOTE method handles the over-fitting issue by adding new minority observations based on the calculated linear interpolations of one of its K-nearest neighbors[5], and the ADASYN method is the SMOTE method with more variance added to the new minority class [6]. Second, after testing the performance of each resampled data, we designed two new methods of over-sampling:

a. Modify the SMOTE method by replacing the embedded KNN algorithm with the K-Means algorithm. We will apply the K-Means algorithm to cluster the features and then find their centroids. we will make duplicates of the minority class by adding noises, which is generated by python random number generator from the continuous uniform distribution to the centroid.

b. Apply the Autoencoder clustering technique to cluster the data to distinct patterns first, and then duplicate data points based on the centroid of each cluster. The autoencoder algorithm has an asymmetric structure. It compresses the input in its encoder part, and after processing the hidden layers, the decoder will decompress the dataset to an output that has the same size as the input dataset. In this project, we will build cluster layers to be our hidden layers, then stack the clustering layer after the pre-trained encoder to form the clustering model. Therefore, we can get a similar-sized output dataset with the assigned clusters to each observation. The coordinates of centroids could be calculated by the formula for each x and y (PCA components in this project) [8]:

$$C_x = \frac{\sum_{i=1}^{n} x_i}{n},\ C_y = \frac{\sum_{i=1}^{n} y_i}{n}$$

After getting centroids from the deep clustering, we will then generate random points using python random generator from the default continuous uniform distribution inside the circle that is been drawn with centroid as the center and the maximum distance between the points and the centroid within each cluster. We will get a space area by drawing the circle with centroid as its center and the maximum Euclidean distance of the points within the cluster to the centroid as its radius. We will place the center of this circle to the origin Then by randomly selecting an angle and a radius with python random generator with the range of 0 to 2Pi for the angle and 0 to the maximum radius, we will get a new data point from doing trigonometric calculations to find an x-coordinate and y-coordinate respect to the angle and the radius. By repeating this process for desired times for each cluster, we will get an over-sampled dataset.

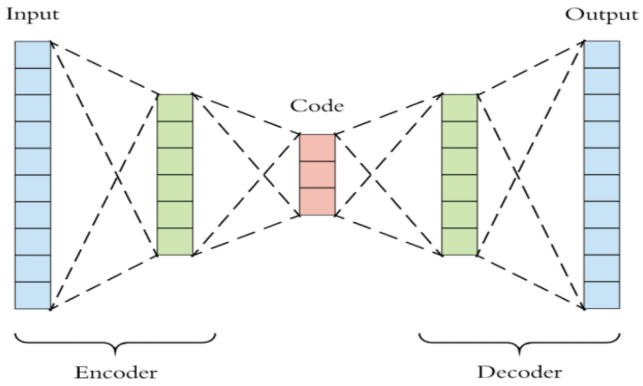

Figure 1. An Autoencoder structure: it contains input layers as the encoder, output layers as the decoder, and hidden layers [9]

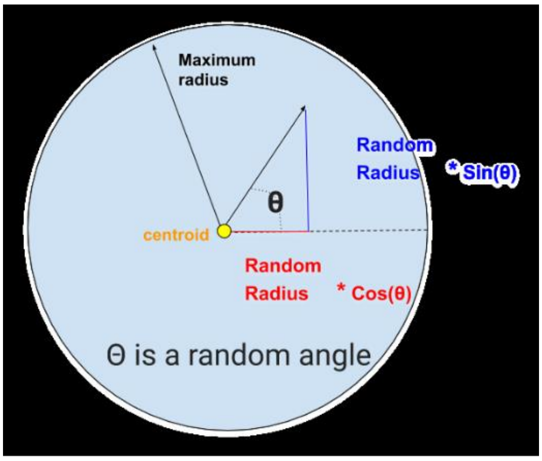

Figure 2. Getting a random point inside the circle with centroid as the center and maximum distance as radius

The reason to modify the over-sampling method is that in general, the over-sampling methods perform better than the under-sampling methods because they keep all the information in the training dataset. With under-sampling, we drop a lot of information. Even if this dropped information belongs to the majority class, it is useful information for a modeling algorithm. If we have a larger dataset, under-sampling may perform better because over-fitting is a potential issue of the over-sampling methods, and then reduce the model performance on the test dataset. In this project, since our subsample is small, it is better to resample it with over-sampling methods.

3.2. Machine Learning Algorithms

In addition to comparing the machine learning algorithms in the work of Doshi et al. (2018) , we are going to add the Gradient Boosting Tree algorithm and the Extreme Boosting algorithm to the resampled data. The gradient boosting tree (GBT) algorithm builds trees one at a time, where each new tree helps to correct errors made by the previously trained trees, and the Extreme Boosting (XGBoost) algorithm optimizes the GBT through parallel processing, tree-pruning, regularization, and cross-validation to avoid overfitting and reduce bias.

3.3. Data Information

The dataset has been used in this paper is from Doshi et al. We simply borrowed their feature selection methods since our goal is to compare the effect of the resampling methods. In the dataset, there are two classes of the labels we are interested in: normal traffic (label=0) and attack traffic (label=1).

Features by categories:

1. Length: In this paper, the length feature is the packet size. The normal packet size is significantly different from the attack packet size since the DoS attack stream would be greedy to open as many connections as possible and hence to make the packet size as small as possible[2].

2. Inter packet interval: According to Doshi et al, the time intervals between packets may reveal the difference between normal and attack streams[2].

3. Protocols. There are four indicator features of protocols: is_TCP, is_UDP, is_HTTP, and is_Other.

4. Bandwidth.

5. IP destination counts and derivatives with respect to a 10-second time window: The count of unique destination and derivative of this number with respect to the time interval of a single IoT device.

In this project, since most of the resampling methods apply distance-based algorithms (KNN, K means), we applied the Principal Component Analysis (PCA) with two components to the eleven features to put them into a 2-D space, so that we would be able to apply Euclidean distance to get centroids within classes or clusters.

4. RESULT AND DISCUSSION

In this section, a subsample with sample size equals 5000 is been drawn from the original dataset due to the extreme large computation expense to the half-million observations. Precision-Recall Curve(PR curve) is used to evaluate the imbalanced data since Davis et al.suggest that the PR curve is more appropriate to evaluate the models with imbalanced data [8]. The Precision is directly influenced by class imbalance so the Precision-recall curves are better to highlight differences between models for highly imbalanced data sets.

4.1. The Figures Below Are the PR Curve of The Neural Networks of The Original Dataset

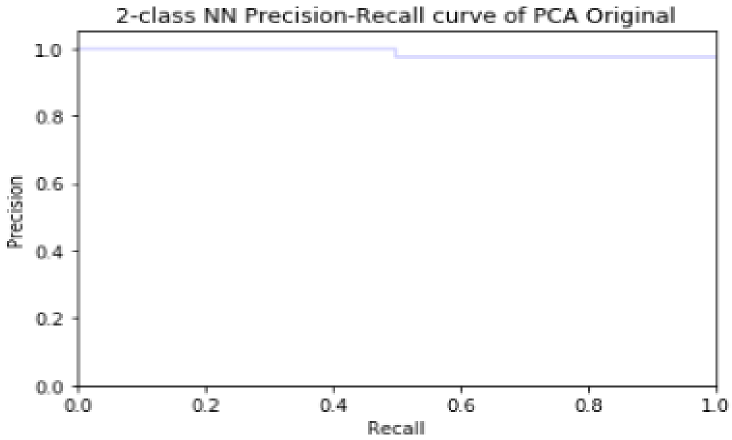

Figure 3. The PR curve of the original dataset

Since the data is now balanced after applying the re-sampling methods, we illustrate the accuracy rate of each Neural Network with different re-sampling methods.

Table 1. Under-sampling data Accuracy of each Neural Network (Sample size = 5000)

Method	Random	Cluster centroids	Tomek links	Neighborhood cleaning rule	Near-miss
Accuracy	0.781	0.813	0.969	0.972	0.953

Table 2. Over-sampling data Accuracy of each Neural Network (Sample size = 5000)

Method	Random	SMOTE	ADASYN	K-means SMOTE	Autoencoder Clustering
Accuracy	0.888	0.888	0.781	0.920	0.949

4.2. The Table Below Shows the Recall Rate of the Original Dataset Applying Multiple Machine Learning Algorithms

Table 3. Recall rate of each Machine learning model without resampling (Sample size = 5000)

Method	Logistic Reg	KNN	SVC	DT	RF	GBT	XGB
Recall rate	1.0	0.999	1.0	0.998	0.999	1.0	0.860

The metrics were extremely high, but these rates are suspicious because some of them have 0 for precision rate and exactly one for recall rate. In addition to the models applied to the work of Doshi et al, we used the grid search algorithms to GBT and XGBoost methods with multiple learning rates and estimators and illustrated the best accuracy below. 10-fold cross-validation is applied to the subsample to avoid overfitting and reduce the bias.

Table 4. The accuracy of the five under-sampling subsamples after applying machine learning and deep learning algorithms.

Methods	Random	Cluster centroids	Tomek links	Neighborhood cleaning rule	Near-miss
Logistic Reg	0.773	0.836	0.930	0.938	0.758
KNN	0.875	0.813	0.984	0.980	0.953
SVC	0.773	0.843	0.930	0.938	0.781
DT	0.938	0.891	0.991	0.991	0.977
RF	0.969	0.875	0.991	0.992	0.961
GBT(best)	0.960	1.0	1.0	1.0	1.0
XGBoost(best)	0.832	0.800	0.778	0.807	1.0

Table 5. The accuracy of the five over-sampling subsamples after applied the machine learning and deep learning algorithms

Methods	Random	SMOTE	ADASYN	K-means SMOTE	Autoencoder clustering
Logistic Reg	0.771	0.751	0.736	0.828	0.847
KNN	0.998	0.992	0.994	0.964	0.971
SVC	0.774	0.755	0.730	0.820	0.839
DT	0.998	0.992	0.993	0.971	0.975
RF	0.998	0.996	0.992	0.981	0.986
GBT(best)	0.998	0.997	0.996	0.981	0.982
XGBoost(best)	0.984	0.909	0.881	0.831	0.886

5. CONCLUSION

To conclude, the Neighborhood cleaning rule method of undersampling outperforms in all of the undersampling methods. The Autoencoder clustering method performs better than the other over-sampling methods. In this case, even the under-sampling methods have a better performance than the over-sampling methods from the metrics, it is highly dangerous to use under-sampling methods with a small size dataset. Since the over-sampling method, Autoencoder clustering method has a comparatively good performance, we would recommend applying this method in the future datasets.

We would suggest using the Neural Networks and the GBT models for future model fitting and prediction since they have higher accuracy in general, and they are more stable to the dataset. With a small 5000-observation sample, the ways of splitting the dataset would change the accuracy of some of the models, but the tree-based models and the Neural Networks are more stable to different seeds of the train-test split process. The overall recall rates or accuracy rates are notably high in this project. One reason is that when doing feature engineering, the selected features are been mathematically transformed and are assumed to be influential to the labels.

REFERENCES

[1] Columbus, L. (2016). Roundup of internet of things forecasts and market estimates, 2016. Forbes Magazine, New York, NY USA.

[2] Al-Garadi, M. A., Mohamed, A., Al-Ali, A., Du, X., & Guizani, M. (2018). A survey of machine and deep learning methods for internet of things (IoT) security. arXiv preprint arXiv:1807.11023.

[3] Doshi, R., Apthorpe, N., & Feamster, N. (2018, May). Machine learning ddos detection for consumer internet of things devices. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 29-35). IEEE.

[4] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.

[5] NV Chawla, KW Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Re- search (JAIR), 16:321–357, 2002.

[6] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 1322-1328). IEEE.

[7] Lee, J., & Lee, J. (2014). K-means clustering based SVM ensemble methods for imbalanced data problem. Paper presented at the 614-617. doi:10.1109/SCIS-ISIS.2014.7044861

[8] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.

[9] Stewart, M. (2019). Comprehensive Introduction to Autoencoders. Retrieved 24 November 2019, from https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368