BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Datasets for Network Security

Jiří Setinský*

Abstract

In the area of network security, we use machine learning techniques to detect anomalies and malware. To get a successful classifier, we need a good quality dataset. In this paper, a quality dataset is obtained using cluster analysis. Cluster analysis makes us possible to filter out only the data that are important to train the final classifier. The approach allows us to efficiently reduce the dataset while enriching it with missing data. The result of the work is an efficient unification of two datasets, where we achieve better results than simply merging and random subsampling. We succeeded in improving the classifier accuracy from 69 % to 93 % with a big reduction of the training dataset. In this way, we can keep the dataset up-to-date to reflect real network traffic, resulting in a more accurate classifier.

*xsetin00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

An accurate classifier is dependent on a quality training dataset. There is a large amount of data floating on the network. The goal is to create a dataset that reflects real traffic while having a compact size.

We want to remove redundant data from the dataset and enrich it with new data that will contribute to higher classifier accuracy. Two datasets are merged, the first dataset is used to train the current classifier. The second dataset contains data captured from the network. The merged dataset is used to train the classifier, which is tested on benchmark data containing disjunctive sets from both datasets.

Existing solutions for reducing or augmenting datasets are written and implemented within the Sklearn framework imbalanced-learn [1]. For example, the framework implements methods for reduction based on the KNN algorithm such as NearMiss [2], TomekLinks [3] or AllKNN [4]. Unlike the proposed solution, the methods work with all classes simultaneously and not all methods allow setting the number of samples for each class. For augmentation, ADASYN [5] or SMOTE [6] methods are implemented, which are methods for generating synthetic data. The proposed framework requires only new data regardless of its origin.

The presented solution uses a clustering algorithm to analyze individual classes. Each class is divided into clusters and a score is computed for each cluster, which serves as a weight for the data to be extracted during the modification process.

The framework is able to efficiently reduce the dataset with minimal loss in model accuracy. Furthermore, it is possible to merge multiple datasets to create a compact dataset with potentially better accuracy across all datasets.

2. Framework

The analysis of the dataset is illustrated in Figure 1. The dataset is split into individual clusters for each class independently by the clustering algorithm (for example k-means). The number of clusters is automatically estimated using the SSE (Sum of Squares Errors) ratio between the K parameters. For each cluster, a set of statistical metrics (mean difference, correlation) is computed. Based on the metrics and the accuracy of the classifier over the cluster, a score is computed (see 2.1). The score value is normalized within the report and will serve us in the modification phase. The score gives us the complexity factor of the model to classify a given cluster.

2.1 Report computation

The calculation of the score is based on the following metrics:

- **Average difference** ($mean_k$): For each feature in

the cluster, the relative deviation from the mean of the entire dataset is calculated. Metric expresses the average of the deviations of all features.

- **Extreme difference** ($extreme_k$): We calculate the proportion of attributes that have relative deviations from the mean greater than 50%. The metric reveals how many attributes differ from the rest of the data.
- **Classifier accuracy deviation** (ad_k): The main indicator in calculating the score is the deviation of the cluster accuracy from the overall accuracy. The deviation is expressed in relative terms. To ensure a higher degree of fluctuation in the accuracy of individual clusters, the classifier used for analysis is ideally trained on a different subset of the data, or training is performed on a very small fraction of the analyzed data.
- **Difference of adversarial attributes** ($contra_k$): Two attributes can be considered as opposites if they show a negative correlation. To illustrate, consider an attribute f_1 that is correlated with a positive class and another attribute f_2 that is correlated with a negative class. The relationship between these attributes should be negatively correlated. If we find that the two attributes are negatively correlated, we look at their mean deviations. Normally, mean deviations should be opposite, but if the deviations are similar, it may tell us that the cluster may be hard to classify.
- **Similarity to another class** (sim_k): Attributes are found that show higher correlation with each class of the dataset. These attributes are then examined for each cluster. The goal is to find clusters that can be similar in attribute values to another class. An attribute correlating with a positive class will have an above average cluster mean deviation in the negative class that overlaps with the positive class. The value of the metric can then identify clusters that tend to overlap with another class.

The total score $score_k$ of the cluster is calculated as the average of all the metrics mentioned using equation (1). All metrics have values in the interval $\langle 0, 1 \rangle$. The values are directly proportional to the complexity of the cluster classification. The moment a metric is constant or achieves a negative correlation with respect to the ad_k metric, it is dropped from the calculation. The resulting score is normalized to the interval $\langle 0.5, 1 \rangle$ using equation (2). The lower bound of 0.5 was chosen to ensure that a given cluster does not completely disappear during modification. If the cluster achieves a normalized score of 1, it represents the data on which the classifier achieves the relative lowest accuracy. The consequence is that the data is preserved in the largest proportion during reduction.

The form of the resulting report for each class and dataset is visualized by Figure 3. The obtained report is then used as a basis for the modification phase. The reduction and augmentation of the data is done in proportion to the individual report weights. The input parameter of the dataset modification is the resulting number of samples of each class or the proportion of data we want to reduce or select for augmentation. The process of reduction and augmentation is very similar. In both cases, there is a weighted subsampling according to the report for the modified dataset. The old dataset is subject to reduction and the relevant data is selected from the new dataset. The modification process is done for each class separately and further we can determine the ratio of reduced and new data in order to obtain a weighted union of the resulting dataset, which will enable the model to perform better after training. The goal of the new data is to bring new objects into the dataset that will provide new information for subsequent model training.

3. Results and Conclusions

In the beginning there were two data sets. The old dataset had a accuracy of 69.2 % Table 3 and the new dataset had a accuracy of 85.4 % Table 4 on the test set. None of the datasets were balanced. The results of merging using the proposed framework were compared with the primitive merging method followed by random subsampling. The merged datasets were always created in such a way that the classes were balanced. For the first dataset created, the size of the old dataset (1,581,269) was maintained. The random subsampling approach achieved a acccuracy of 93.01 % Table 5. Merging the datasets using cluster analysis yielded a acccuracy of 93.27 % Table 6. The second dataset created was reduced by 90 % (158,125). Random subsampling achieved a accuracy of 92.98 % Table 1 and the clustering-based approach achieved a accuracy of 93.97 % Table 2.

A framework for efficient dataset creation has been proposed. The results show that the framework is able to produce a dataset that achieves better results than using random merging. It even manages to improve the accuracy of the model despite a large reduction ratio, thus minimizing the computational effort of training.

Acknowledgements

I would like to thank my supervisor Martin Žádník and Peter Tisovčík.

References

[1] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[2] Inderjeet Mani and I Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, pages 1–7. ICML, 2003.

[3] Tomek Ivan. Two modifications of cnn. *IEEE transactions on Systems, Man and Communications, SMC*, 6:769–772, 1976.

[4] Ivan Tomek. An experiment with the edited nearest-nieghbor rule. 1976.

[5] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.