A Novel Hybrid Data Reduction Strategy and Its Application to Intrusion Detection

Vitali Herrera-Semenets(✉), Osvaldo Andrés Pérez-García, Andrés Gago-Alonso, and Raudel Hernández-León

Advanced Technologies Application Center (CENATAV), 7a # 21406, Rpto. Siboney, Playa, 12200, Havana, Cuba
{vherrera,osvaldo.perez,agago,rhernandez}@cenatav.co.cu

Abstract. The presence of useless information and the huge amount of data generated by telecommunication services can affect the efficiency of traditional Intrusion Detection Systems (IDSs). This fact encourage the development of data preprocessing strategies for improving the efficiency of IDSs. On the other hand, improving such efficiency relying on the data reduction strategies, without affecting the quality of the reduced dataset (i.e. keeping the accuracy during the classification process), represents a challenge. Also, the runtime of commonly used strategies is usually high. In this paper, a novel hybrid data reduction strategy is presented. The proposed strategy reduces the number of features and instances in the training collection without greatly affecting the quality of the reduced dataset. In addition, it improves the efficiency of the classification process. Finally, our proposal is favorably compared with other hybrid data reduction strategies.

Keywords: Data mining · Data reduction · Instance selection
Feature selection

1 Introduction

Nowadays, the volume of data generated from using telecommunication services is considerably large, which causes Big Data challenges in the network traffic [1]. For example, AT&T long distance customers alone generate over 300 million records (instances) per day [2]. The presence of non-relevant information or noise in the data could affect the performance of the learning methods used to detect events representing attacks. Additionally, such events are executed by malicious users known as intruders, causing millions in losses and damaging the prestige of the affected companies. In order to analyze such volume of data and quickly detect which events are associated to an attack, it is necessary to apply intrusion detection techniques.

In literature, there are two main approaches based on data mining for the intrusion detection problem: (1) supervised approach, which requires a labeled training set T [3], and (2) unsupervised approach, where previous knowledge

M. Mendoza and S. Velastín (Eds.): CIARP 2017, LNCS 10657, pp. 289–297, 2018.
https://doi.org/10.1007/978-3-319-75193-1_35

is not required [4]. In this paper, we propose a novel hybrid data reduction strategy (following the supervised approach), called HDR, that combines both feature selection and instance selection to obtain a reduced training set $S \subset T$, providing high efficiency without affecting the quality of the reduced dataset too much. We evaluate the quality of the data during the classification process. For this, we use three standard measures: Accuracy, Recall and False Positive Rate.

2 Related Work

Data can be reduced in terms of the number of rows (instances) or in terms of the number of columns (features) [5]. In general, three main approaches have been proposed: (1) feature selection based, (2) instance selection based and hybrid, where feature selection and instance selection are combined [6].

The feature selection algorithms look up the most relevant features of the dataset. In this way, only a subset of features from the underlying data is used in the analytical process. On the other hand, the instance selection algorithms obtain a reduced subset of instances S from the original training set T, so that S does not contain superfluous instances. Instance selection approach can either start with $S = \emptyset$ (incremental methods) or $S = T$ (decremental methods) [7]. Incremental methods obtain S by selecting instances from T [8], while decremental ones obtain S by deleting instances from T [9].

According to the strategy used for selecting instances, the algorithms can be divided into two groups: Wrapper and Filter [7]. In Wrapper algorithms, the classifier is used in the selection process, the instances which do not affect the classification accuracy are removed from T. On the other hand, Filter algorithms are independent from the classifiers and the selection criterion is based on different heuristics.

In [6], a hybrid method for intrusion detection is proposed. In this case, the feature selection process is performed by the OneR [10] algorithm. Then, an expensive clustering algorithm, called Affinity Propagation [11], is used for instance selection process. Also, in order to improve the performance and scalability of the method, they implemented a distributed solution using MapReduce.

However, the main problem of the instance selection methods is the high runtime when large datasets are processed, which makes unfeasible their application in some cases, and directly affects the performance of the hybrid approaches.

3 Our Proposal

In this section, we introduce the Hybrid Data Reduction (HDR) strategy. The HDR strategy has three main phases: (1) feature selection phase, (2) relabeling phase and (3) instance reduction phase.

Feature Selection Phase
Most of the reported works in these scenarios use only one feature selection metric, and do not take advantage of the possibilities that the combination

of different metrics can offer; since different metrics could measure different information in the features. Therefore, as final result, different features with the same level of data representativeness could be selected. In this sense, our hypothesis is that a better management of these metrics could lead to a better selection of the final set of features. After a study, we determined that the most commonly used measures can be grouped into three categories: entropy based (Information Gain, Gain Ratio, and Symmetric Uncertainty), statistical based (Chi-square), and instance based (Relief and ReliefF) [12].

In our proposal, we use three different algorithms (one representative from each category): ReliefF, Chi-squared Ranking Filter and Information Gain Ranking Filter. The ReliefF algorithm estimates how well a feature can differentiate instances from different classes by searching for the nearest neighbors of the instances from the same and different classes. Chi-squared is a nonparametric statistical measure that estimates the correlation between the distribution of an attribute and the distribution of the class. In the case of Information Gain, it measures the amount of information that a feature can provide about whether an instance belongs to one class or another.

In Algorithms 1 (lines 4–8) and 2, the proposed feature selection strategy is described. Notice that for each feature selection algorithm, the score mean is computed, and the features whose values exceed the score mean are selected. Finally, the union of the three resulting sets is returned.

Algorithm 1. HDR

```
   Input: T: training set
   Output: S: reduced training set
1  S ← T
2  F ← ∅ // selected features set
3  L ← ∅ // generated labels set
   /*Feature selection phase*/
4  F_RF ← Selector (ReliefF_FS (S))
5  F_CHI ← Selector (Chi_Square_FS (S))
6  F_IG ← Selector (InfoGain_FS (S))
7  F ← F_UFS ∪ F_CHI ∪ F_IG
8  S ← Dimensionality-Reduction (F, S)
   /*Relabeling phase*/
9  L ← Label-Generation (S)
10 S ← Relabeling (S, L)
   /*Instance reduction phase*/
11 S ← Duplicated-Removing (S)
12 return S
```

Algorithm 2. Selector

```
   Input: P_A: Set of scores
          assigned by A
   Output: F_A: Set of features
           selected from A
1  F_A ← ∅ //
2  p̄_A ← Mean-Score(P_A)
3  foreach p_f ∈ P_A do
4  |   if p_f > p̄_A then
5  |   |   F_A ← F_A ∪ {f}
6  |   end
7  end
8  return F_A
```

Relabeling Phase

After reducing the number of features in S, the relabeling process is carried out to generate new labels for the selected features values. In order to gain efficiency, we use the k-means algorithm to generate the labels during the $Label-Generation$ function (see Algorithm 1, line 9). The purpose of applying a clustering method as part of the relabeling process is to search for similar values and group them into clusters.

For each selected numerical feature f_i, the k-means algorithm is executed over the set of values taken by f_i in S, denoted by V_i. Each of the obtained clusters contains a range of numerical values, which are represented by a unique numerical label (see Algorithm 1, line 10). The use of these clusters allow us to cover feature values that do not exist in S and are included in the classification stage.

For example, suppose that in the training phase a feature f_1 takes values in the set $V_1 = \{0, 0.2, 0.3, 0.6, 0.7, 1.0\}$, and during the relabeling process a cluster $c_1 = [0, 0.2, 0.3, 0.6]$ and $c_2 = [0.7, 1.0]$ are obtained. Then, in the classification stage, it is required to classify a new transaction, in which the feature f_1 takes the value $0.9 \notin V_1$, but 0.9 falls within the range of c_2, therefore it can be classified.

Instance Reduction Phase
Finally, the instance reduction phase is performed over the relabeled training collection. Here, $Duplicated-Removing$ function, as its name suggests, removes duplicate instances from S (see Algorithm 1, line 11). Notice that an instance is a duplicate instance if there is at least another instance with the same feature values and class. The result is a reduced training collection. This collection is used by a classifier to build a classification model.

4 Experimental Results

In this section, we aim to evaluate the novel hybrid data reduction strategy introduced in this paper; comparing its efficiency and the quality of the reduced dataset against the hybrid algorithm proposed by Chen *et al.* [6].

The experiments were conducted using two different datasets: KDD'99 [13] and CDMC 2013 [14]. The KDD'99 dataset has been widely used for testing network intrusion detection approaches, and it is considered a benchmark dataset. This dataset provides instances, each one containing 41 features out of which 9 are discrete and 32 are continuous. The training set consists of 494021 instances, while the testing set contains 311029 instances. In our experiments, we classify all the instances into two types, "normal" or "attack". In the case of CDMC 2013, it is an intrusion detection dataset collected from a real intrusion detection system. Each instance of this dataset contains 7 numerical features and a label indicating whether an instance is related with an attack or not. CDMC 2013 consists of 71758 "normal" labeled instances, and 6201 "attack" labeled instances. Without loss of generality, the dataset was splitted into two subsets: one for training (40000 instances), and another for testing (37959 instances).

Our experiments were performed on a PC equipped with 2.5 GHz Intel Core 2 Quad CPU and 4.00 GB DDR2 of RAM, running Windows 8. In Chen *et al.* [6], the KDD'99 dataset was evaluated using a specific configuration, which was the same used in our experiments. On the other hand, the CDMC 2013 dataset was not evaluated in [6], therefore, to establish a fair comparison, we adjusted the configuration of its proposal in order to obtain a reduced dataset with a similar

size to the one obtained by HDR. We decided to test our strategy using different k values for the relabeling process.

In Fig. 1 we show the runtime of Chen *et al.* [6] method and HDR for different volumes of KDD'99 dataset. Notice that the execution time of HDR and the k value are directly proportional. This is because the k intervals in which a feature value is divided may have different sizes, and we have to check all of them (in the worst case) to assign the new labels. According to the performance shown in Fig. 1, we can conclude that regardless of the k value used, the HDR strategy is more efficient than the proposal of Chen *et al.* [6], and the HDR algorithm scales linearly regarding to k. Because the size of KDD'99 dataset is more than five times the size of CDMC 2013 dataset, and due to space reasons, we consider that it is enough to show this experiment using only the KDD'99 dataset.

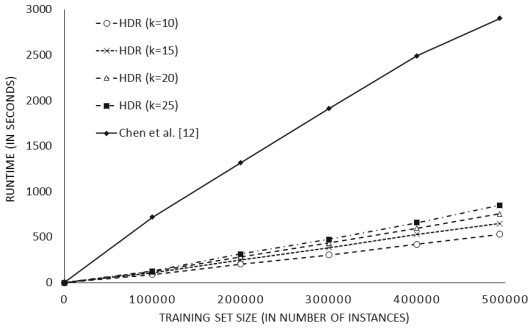

Fig. 1. Performances of Chen *et al.* [6] and HDR for different volumes of KDD'99 training dataset.

After the reduced training set S is obtained, two traditional classifiers are used in our experiments to evaluate the quality of our data reduction strategy, such classifiers are KNN and SVM. Both classifiers are commonly accepted for classification in intrusion detection scenario [15,16]. Similar to Chen *et al.* [6], for KNN, we set the parameter $k = 1$. To evaluate the quality of the reduction process, we used three standard measures: Accuracy (see Eq. 1), Recall (see Eq. 2) and False Positive Rate (FPR) (see Eq. 3).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100, \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \cdot 100, \tag{2}$$

$$FPR = \frac{FP}{FP + TN} \cdot 100, \tag{3}$$

where TP, TN, FP and FN represents the true positive, true negative, false positive and false negative, respectively.

Table 1. Results achieved using KDD'99 dataset.

Classifier	Data reduction strategy	Acc	Recall	FPR	Instances	Features
SVM	Baseline	92.30	90.8	1.7	494021	41
SVM	Chen *et al.* [6]	90.05	87.9	2.1	63234	17
SVM	HDR (k = 10)	87.03	86.1	2.7	22822	17
SVM	HDR (k = 15)	88.31	86.9	2.4	38679	17
SVM	HDR (k = 20)	88.83	87.3	2.1	55999	17
SVM	HDR (k = 25)	90.26	88.5	2.0	60540	17
KNN	Baseline	92.81	91.2	0.7	494021	41
KNN	Chen *et al.* [6]	89.51	88.8	1.1	63234	17
KNN	HDR (k = 10)	87.12	86.7	2.2	22822	17
KNN	HDR (k = 15)	87.93	87.6	1.6	38679	17
KNN	HDR (k = 20)	89.10	88.4	1.2	55999	17
KNN	HDR (k = 25)	89.97	89.1	1.0	60540	17

In Tables 1 and 2, we show the values of the three quality measures mentioned above together with the number of instances and features, for HDR and Chen *et al.* [6] proposal, over the two datasets evaluated. Taking into account that in [6] it is necessary to predefine the number of features to be selected, in order to make a fair comparison, we decided that it will be the same as the amount of features selected by our proposal. Additionally, we include the results using the original training set without any reduction (baseline).

From Tables 1 and 2, it can be seen that the performance of HDR improves as the k values increase. In Table 1, we can see that for $k = 25$ the HDR strategy reaches better accuracy with less instances than Chen *et al.* [6] proposal. Additionally, HDR strategy obtains better results with respect to the recall and the FPR quality measures. Similar results are shown in Table 2 for $k = 20$ in the CDMC 2013 dataset.

Note that despite the considerable reduction of the dataset, the quality measures obtained using HDR with $k = 25$ do not vary greatly regarding to those obtained using the baseline. This shows that the quality of the reduced data was not affected too much. In addition, according to HDR behavior during the experiments, if a higher k value is defined, the dataset is reduced to a lesser extent, but the quality of the reduced data is improved, which is clearly shown by the improvement in the quality measures.

Our last experiment is to demonstrate how our proposal contributes to the classifiers efficiency during the classification process. For space reasons, and because KNN is much slower than SVM, we decided to show only the results achieved by KNN. Such results are shown in Fig. 2 for both KDD'99 and CDMC 2013 datasets respectively.

As it can be seen in Fig. 2(a), by using KNN with HDR it was possible to reduce the time consumed by 36 % regarding to the baseline, which means that KNN using HDR with $k = 25$ is 15807 s faster than KNN using the original

Table 2. Results achieved using CDMC 2013 dataset.

Classifier	Data reduction strategy	Acc	Recall	FPR	Instances	Features
SVM	Baseline	96.42	95.9	0.2	40000	7
SVM	Chen *et al.* [6]	94.21	93.0	1.2	984	4
SVM	HDR (k = 10)	91.47	91.3	2.1	239	4
SVM	HDR (k = 15)	94.16	92.7	1.5	500	4
SVM	HDR (k = 20)	94.76	93.1	0.9	825	4
SVM	HDR (k = 25)	95.41	94.0	0.4	1086	4
KNN	Baseline	96.57	96.1	0.1	40000	7
KNN	Chen *et al.* [6]	94.78	93.9	1.0	984	4
KNN	HDR (k = 10)	92.25	91.9	1.8	239	4
KNN	HDR (k = 15)	94.72	93.3	1.1	500	4
KNN	HDR (k = 20)	95.11	94.2	0.6	825	4
KNN	HDR (k = 25)	95.87	94.9	0.3	1086	4

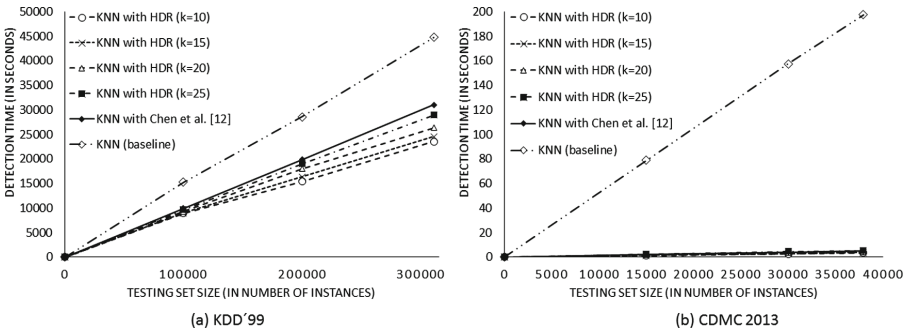

Fig. 2. Comparative results of classification time using KNN for KDD'99 and CDMC 2013 testing sets.

dataset (baseline). In addition, a better efficiency than the Chen *et al.* [6] proposal was achieved during the classification process using HDR.

On the other hand, for CDMC 2013 dataset, we also achieved improvements in efficiency during the classification process (see Fig. 2(b)). This time, the KNN classifier using HDR with $k = 20$ achieved better efficiency than KNN using Chen *et al.* [6] proposal. In this sense, KNN using HDR with $k = 20$, it was able to reduce the detection time by 98 % regarding to the baseline, being in turn 0.8 % faster than KNN using Chen *et al.* [6] proposal.

As we saw in this experiment, HDR improves the performance of KNN classifier by decreasing the time spent during the classification process. This can be very useful in scenarios such as intrusion detection, where real-time data analysis may be necessary. Our proposal is capable of achieving a better efficiency than the baseline, without greatly affect the quality of the data during the reduction process.

5 Conclusions

In this paper, we present a novel hybrid data reduction strategy composed of three main phases: (1) feature selection phase, (2) relabeling phase and (3) instance reduction phase. The experimental results show that HDR greatly reduces the original dataset, without affecting the quality of the reduced dataset too much. Also, it was able to further reduce the dataset than the other hybrid approach evaluated, and even so, the classifiers using HDR achieved better results. On the other hand, our proposal does not require large computational resources. HDR is able to process large volumes of data with good performance on a standard PC. Improving the performance of the classifiers in terms of the time elapsed during the classification process is another contribution of our proposal, which can be very useful in scenarios where it is necessary to process data in real time. In addition, HDR did not show any significant increase of its running time, which makes it feasible to process large volumes of data, where data reduction is essential. However, an interesting direction for future work is to evaluate our proposal in other scenarios with higher dimensional spaces, in terms of features.

References

1. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. J. Big Data **2**(1), 1–41 (2015)
2. Cortes, C., Pregibon, D.: Signature-based methods for data streams. Data Mining Knowl. Discov. **5**(3), 167–182 (2001)
3. Kotsiantis, S.B.: Supervised machine learning: a review of classification techniques. In: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering, Amsterdam, The Netherlands, pp. 3–24 (2007)
4. Ghahramani, Z.: Unsupervised learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) ML -2003. LNCS (LNAI), vol. 3176, pp. 72–112. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28650-9_5
5. Aggarwal, C.C.: Data Mining: The Textbook. Springer, Heidelberg (2015)
6. Chen, T., Zhang, X., Jin, S., Kim, O.: Efficient classification using parallel and scalable compressed model and its application on intrusion detection. Expert Syst. Appl. **41**(13), 5972–5983 (2014)
7. Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artif. Intell. Rev. **34**(2), 133–143 (2010)
8. Chou, C.H., Kuo, B.H. and Chang, F.: The generalized condensed nearest neighbor rule as a data reduction method. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 2, pp. 556–559 (2006)
9. Wilson, D.R., Martínez, T.R.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**(3), 257–286 (2000)
10. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Mach. Learn. **11**(1), 63–90 (1993)
11. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)

12. Liu, W., Liu, S., Gu, Q., Chen, J., Chen, X., Chen, D.: Empirical studies of a two-stage data preprocessing approach for software fault prediction. IEEE Trans. Reliab. **65**(1), 38–53 (2016)
13. KDDCup 1999: Computer network intrusion detection. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 25 Feb 2017
14. Song, J.: CDMC2013 intrusion detection dataset. Department of Science & Technology Security, Korea Institute of Science and Technology Information (KISTI) (2013)
15. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst. Appl. **38**(1), 306–313 (2011)
16. Aburomman, A.A., Reaz, M.B.I.: A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl. Soft Comput. **38**, 360–372 (2016)