CHAPTER 6
NETWORK INTRUSION DETECTION USING GSVM-REPETITIVE UNDER SAMPLING

6.1 INTRODUCTION

Highly imbalanced classification is important and increasingly common with emergence of new machine intelligence application domains. In order to solve this challenging class imbalance problem, a novel Granular Support Vector Machines - Repetitive Under sampling algorithm (GSVM-RU) is designed in this work. GSVM-RU creatively utilizes the Support Vector Machines (SVM) for itself under sampling to minimize the negative effect of information loss while maximizing the positive effect of data cleaning in the under sampling process. Consequently, an accurate and fast classifier can be modeled.

Many methods have been proposed for imbalanced classification and some good results have been reported. These methods can be categorized into three different classes: weight adjusting, boundary alignment, and sampling. Sampling can be further categorized into two subclasses: over sampling the minority class, or under sampling the majority class. Most of current methods use decision trees as the basic classifier. Although there are already some works on SVM for imbalanced classification [Japkowicz and Stephen, 2002], the application of SVM for under sampling is still unexplored. Because SVM decides the class label of a sample based only on Support Vectors (SV), which are the training samples close to the decision boundary. The modeling effectiveness and efficiency may be improved for imbalanced classification in a SV based under sampling way.
From the viewpoint of granular computing, most of the current methods, including cost sensitive boosting, bagging, under sampling the majority class, or oversampling the minority class, can be actually viewed as granular computing based because all of them try to form multiple information granules with suitable numbers and suitable sizes.

Sampling strategies, such as over sampling and under sampling, are extremely popular in tackling the problem of class imbalance. That is, either the minority class is over sampled or majority class is under sampled or some combination of the two is deployed. In this work, we focus on learning Support Vector Machines with different sampling techniques. We focus on comparing the methodologies on the aspects of effectiveness and efficiency. While, effectiveness and efficiency can be application dependent, in this work we consider them as follows: Effectiveness means the ability of a model to accurately classify unknown samples, in terms of some metric. Efficiency means the speed to use a model to classify unknown samples.

SVM embodies the structural risk minimization principle to minimize an upper bound on the expected risk. Because structural risk is a reasonable trade-off between the training error and the modeling complication, SVM has a superior generalization capability. Geometrically, the SVM modeling algorithm works by constructing a separating hyper plane with the maximal margin. Compared with other standard classifiers, SVM is more accurate on moderately imbalanced data. The reason is only Support Vectors (SV) are used for classification and many majority samples far from the decision boundary can be removed without affecting classification.

The SVM classifier can be sensitive to high class imbalance, resulting in a drop in the classification performance on the positive class. It is prone to generate
a classifier that has a strong estimation bias towards the majority class, resulting in a large number of false negatives. There have been works in improving the classification performance of SVM on unbalanced datasets. However, they do not address efficiency very well, and depending on the strategy for countering imbalance, they can take a longer time for classification than a standard SVM. Also, SVM can be slow for classification on large datasets. The speed of SVM classification depends on the number of SVs. For a new sample $X$, $K(X, SV)$, the similarity between $X$ and $SV$, is calculated for each $SV$. Then it is classified using the sum of these kernel values and a bias. To speed up SVM classification, one method is to decrease the number of $SV$s.

GSVM-RU can improve classification performance by extracting informative samples that are essential for classification and eliminating a large amount of redundant or even noisy samples. Besides, GSVM-RU also proposes other SVM modeling methods that overweight the minority class, oversample the minority class, or undersample the majority class.

6.2 BASIC ISSUES OF GRANULAR COMPUTING

Granular computing is based on two related issues: granulation and computation. The former deals with the construction, interpretation, and representation of the three basic components, and the latter deals with the computing and reasoning with granules and granular structures.

6.2.1 Granulation

Granulation involves the construction of the three basic components, granules, granulated views and hierarchies. Two basic operations are the top down decomposition of large granules to smaller granules, or the bottom-up combination of smaller granules into larger granules. A family of granules containing every
object in the universe is called a granulated view of the universe. A granulated view may consist of a family of either disjoint or overlapping granules.

Granulation criteria: A granulation criterion deals with the semantic issues and addresses the question of why two objects are put into the same granule. It is domain specific and relies on the available knowledge. In many situations, objects are usually grouped together based on their relationships, such as indistinguishability, similarity, proximity, or functionality. One needs to build models to provide both semantical and operational interpretations of these notions. They enable us to formally and precisely define various notions involved, and to systematically study the meanings and rationale of a granulation criterion.

Granulation methods: From the algorithmic aspect, a granulation method addresses the problem of how to put two objects into the same granule. It is necessary to develop algorithms for constructing granules and granulated views efficiently based on a granulation criterion.

Representation/description: The next issue is the interpretation of the results of a granulation method, i.e., the granular structures. Once constructed, it is necessary to describe, to name and to label granules using certain languages. One may assign a name to a granule such that an element in the granule is an instance of the named category. One may also provide a formal description of objects in the same granule. By pooling the representations of granules, one can obtain the overall representation of a granulated view.

Qualitative and quantitative characterization: One can associate quantitative measures to the three components, granules, granulated views, and hierarchies. The measures should reflect and be consistent with the three structures, the internal structure of a granule, the collective structure of a granulated view, and the overall structure of a hierarchy.
6.3 GSVM-REPETITIVE UNDER SAMPLING METHOD

Granular computing represents information in the form of some aggregates called information granules such as subsets, subspaces, classes, or clusters of a universe. It then solves the targeted problem in each information granule [Lin 2000]. The main two principles in granular computing are given below.

- The first principle is divide-and-conquer to split a huge problem into a sequence of granules (Granule Split).
- The second principle is data cleaning to define the suitable size for one granule to comprehend the problem at hand without getting buried in unnecessary details (Granule Shrink).

SVM assumes that only SVs are informative to classification and other samples can be safely removed. However, for highly imbalanced classification, the majority class pushes the ideal decision boundary towards the minority class. To find informative samples, one can conduct cost sensitive learning or over sampling. However, these two “rebalance” strategies increase the number of SVs and hence slow down the classification process. To improve efficiency, it is natural to decrease the size of the training dataset. In this sense, under sampling is by nature more suitable to model an SVM for imbalanced classification than other approaches. However, elimination of some samples from the training dataset may have two effects. Information loss: due to the elimination of informative or useful samples, classification effectiveness is deteriorated; Data cleaning: because of the elimination of irrelevant or redundant or even noisy samples, classification effectiveness is improved.

For a highly imbalanced dataset, there may be many redundant or noisy negative samples. Random under sampling is a common under sampling approach for rebalancing the dataset to achieve better data distribution. However, random under sampling suffers from information loss. Although random under sampling
pushes the learned boundary close to the ideal boundary, the cues about the orientation of the ideal boundary may be lost.

GSVM-RU is targeted directly to utilize SVM itself for under sampling. The idea is based on the well known fact about SVM - only SVs are necessary and other samples can be safely removed without affecting classification. This fact motivates us to explore the possibility to utilize SVM for data cleaning/under sampling. However, because of highly skewed data distribution, the SVM modeled on the original training dataset is prone to classify every sample to be negative. Therefore, a single SVM cannot guarantee to extract all informative samples as SVs. Fortunately, it seems reasonable to assume one single SVM can extract a part of, although not all, informative samples. Under this assumption, multiple information granules with different informative samples can be formed by following granulation operations.

**Step 1:** Firstly, assume that all positive samples are informative to form a positive information granule. Secondly, negative samples are extracted by an SVM as SVs are also possibly informative so that they form a negative information granule. These negative samples are called as Negative Local Support Vectors (NLSVs). These NLSVs are removed from the original training dataset to generate a smaller training dataset, on which a new SVM is modeled to extract another group of NLSVs.

**Step 2:** This process is repeated several times to form multiple negative information granules.

**Step 3:** Then, all other negative samples still remaining in the training dataset are simply discarded.

**Step 4:** An aggregation operation is then executed selectively to aggregate the samples in these negative information granules with all positive samples to complete the under sampling process.
**Step 5**: Finally, an SVM is modeled on the aggregated dataset for classification. GSVM-RU under sampling can still give good cues about the orientation of the ideal boundary, and hence can overcome the shortcomings of random under sampling.

Figure 6.1, sketches the idea of GSVM-RU. For SVM modeling, GSVM-RU adds another hyper parameter $Gr$, the number of negative granules. It seems safe to extract more granules to reduce information loss. However, information contributed by two different granules may be redundant or even noisy to each other. And hence, fewer granules may decrease these redundancy or noise from the final aggregation dataset. In general, if $Gr$ granules are extracted, have $2^{Gr}$ different combinations to build the final aggregation dataset. It is extremely expensive to try all of these combinations. For simplicity and efficiency, revise the preliminary GSVM-RU algorithm and propose to run granulation and aggregation in turns in this work. Firstly, the aggregation dataset is initialized to consist of only positive samples. And the best classification performance is initialized to be the performance of the naive classifier that classifies every sample as negative. When a new negative granule is extracted, the corresponding NLSVs are immediately aggregated into the aggregation dataset.

An SVM is then modeled on this new aggregation dataset. The classification performance must be improved, continued to the next phase to extract another granule. Otherwise the repetitive under sampling process will be stopped and the classifier in the previous phase will be saved for future classification. When a new negative granule is extracted, only negative samples in the latest granule are aggregated into the new aggregation dataset and all samples in old negative granules are discarded. This operation is based on the “boundary push” assumption. If old NLSVs are discarded, the decision boundary is expected to be closer to the ideal one. The repetitive under sampling process is stopped when the new extracted granule cannot further improve classification.
performance. The repetitive under sampling process is stopped when the new extracted granule cannot further improve classification performance if joined with the previous aggregation dataset.

**Figure 6.1 Basic idea of GSVM-RU**

Please purchase PDF Split-Merge on www.verypdf.com to remove this watermark.
Algorithm GSVM-RU

**Initialization**
Given the training samples \( x_i \) \((x_i, y_i) \) \( i = 1 \)

**Step 1: Granulating based on kernel**
Given the number of granulation parameters k and take granulation based on granular dividing algorithm. Obtain the divided granules \( \{x_1, x_2, \ldots, x_k\} \).

**Step 2: Extracting mixed granules**
Set up the threshold parameter of mixed granule support \( I \) and purity \( i \), then take the samples of mixed granule into the set(mixed).

\[
\text{while } (\text{the size of granule } x_i \text{ in set (mixed) is bigger than } \frac{2I}{k}) \\
\{ \\
\quad \text{Delete the mixed } x_i \text{ from set(mixed).} \\
\quad \text{Divide mixed granule } x_i \text{ into sub granules based on granular dividing algorithm.} \\
\quad \text{Add mixed sub granules into the set(mixed).} \\
\}
\]

**Step 3: Training SVM**
Take all the samples of set(mixed) as training samples, and train SVM. Then an initial approximate hyper plane \( f \) is obtained.

**Step 4: Correcting hyper plane**
Set up the threshold parameter \( d' (d > d') \) of hyper plane correction.
Compute the distance from each unity granule super ball to the initial hyper plane.
For each purity granule \( x_i \)
\[
\{ \\
\quad \text{Add all samples } x_{i,j} \text{ in granules } x_i \text{ into the training dataset, if } (d(x_{i,j}, f) < d) \\
\}
\]

**Step 5: Retraining SVM**
Train SVM on the new training samples and obtain the final superior hyper plane \( f' \).
Proposed GSVM-RU Technique

**Step 1:** The raw dataset is taken as input

**Step 2:** That dataset is subjected to preprocessing

**Step 3:** Preprocessing - The symbols in the dataset is converted into numerics

**Step 4:** The dataset is normalized by scaling. It converts numerics into normalized values

**Step 5:** The dataset is split into train and test data

**Step 6:** Training: In this phase the training data are trained using GSVM-RU Classifier. It extracts the granules

**Step 7:** Testing: The test data is classified based on the trained granules

**Step 8:** Then the classified results of normal and abnormal attacks for the protocols is obtained
Figure 6.2 Block Diagram of Network Intrusion Detection using GSVM-RU
6.4 EXPERIMENTAL RESULTS

6.4.1 Performance Evaluation for KDD Cup'99 Data Set

For the KDD Cup'99 and NSL-KDD data sets the detection rates, false alarm rates and accuracy are calculated as follows.

a) Detection Rate Comparison

In this part the detection rate of attacks of the proposed method GSVM-RU is compared with the traditional SVM algorithm.

Table 6.1 Comparison Table for Detection Rate of attacks for SVM and GSVM-RU

<table>
<thead>
<tr>
<th>Attacks/Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>84.14</td>
<td>87.24</td>
<td>84.91</td>
<td>86.88</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>92.47</td>
<td>93.74</td>
<td>91.69</td>
<td>93.87</td>
</tr>
</tbody>
</table>

From the table 6.1, SVM for the four attacks of DoS, Probing, U2R and R2L give values of 84.14, 87.24, 84.91 and 86.88. Where else GSVM-RU gives 92.47, 93.74, 91.69 and 93.87 which are better than SVM detection rate.

![Comparison Graph for Detection Rate of Attacks for SVM and GSVM-RU](image)

*Figure 6.3 Comparison Graph for Detection Rate of Attacks for SVM and GSVM-RU*
From the figure 6.3, it is clear that for the four attacks DoS, Probe, U2R and R2L the detection rate of GSVM-RU gives higher percentage of detection when compared with traditional SVM.

b) False Alarm Rate Comparison

In this part the false alarm rate of attacks of the proposed method GSVM-RU is compared with the traditional SVM algorithm.

<table>
<thead>
<tr>
<th>Attacks/Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>1.92</td>
<td>0.68</td>
<td>1.32</td>
<td>2.56</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>1.32</td>
<td>0.32</td>
<td>0.65</td>
<td>0.68</td>
</tr>
</tbody>
</table>

From the table 6.2, SVM for the four attacks of DoS, Probing, U2R and R2L gives higher alarm rate of 1.92, 0.68, 1.32 and 2.56. For the proposed GSVM-RU, it produces minimum rate of 1.32, 0.32, 0.65 and 0.68 which are better than SVM alarm rate.

![Figure 6.4 Comparison Graph for False Alarm Rate for SVM and GSVM-RU](image-url)
From the figure 6.4, it can be seen that for the four attacks DoS, Probe, U2R and R2L the false alarm rate of GSVM-RU produces minimum value when compared with traditional SVM.

c) Accuracy

In this part the accuracy of the proposed method GSVM-RU is compared with the traditional SVM algorithm.

<table>
<thead>
<tr>
<th>Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>85.3</td>
<td>87.8</td>
<td>84.5</td>
<td>80</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>95</td>
<td>95.9</td>
<td>96</td>
<td>96.5</td>
</tr>
</tbody>
</table>

From the table 6.3, SVM for the four attacks of DoS, Probing, U2R and R2L gives accuracy of 85.3, 87.8, 84.5 and 80. For the proposed KSVM, it gives accuracy of 95, 95.9, 96 and 96.5 which are better than SVM accuracy.

From the figure 6.5, it can be seen that for the four attacks DoS, Probe, U2R and R2L accuracy of GSVM-RU produces higher accuracy when compared with traditional SVM.
6.4.2 Performance Evaluation for NSL-KDD Data Set

For the NSL-KDD dataset, the detection rate of attacks, false alarm rates and accuracy are calculated as previous one.

a) Detection Rate Comparison

In this part the detection rate of attacks of the proposed method GSVM-RU is compared with the traditional SVM algorithm.

Table 6.4 Comparison Table for Detection Rate of Attacks for SVM and GSVM-RU

<table>
<thead>
<tr>
<th>Attacks/Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>87.82</td>
<td>90.47</td>
<td>88.35</td>
<td>85.81</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>94.31</td>
<td>93.52</td>
<td>93.82</td>
<td>93.97</td>
</tr>
</tbody>
</table>
From the table 6.4, SVM for the four attacks of DoS, Probing, U2R and R2L give values of 87.82, 90.47, 88.35 and 85.81. Where else GSVM-RU method gives 94.31, 93.52, 93.82 and 93.97 which are better than SVM method detection rate.

![Graph for Detection Rate of Attacks for SVM and GSVM-RU](image)

**Figure 6.6 Comparison Graph for Detection Rate of Attacks for SVM and GSVM-RU**

From the figure 6.6, it is clear that for the four attacks DoS, Probe, U2R and R2L the detection rate of GSVM-RU method gives higher percentage of detection when compared with traditional SVM method.

b) **False Alarm Rate Comparison**

In this part the false alarm rate of attacks of the proposed method GSVM-RU is compared with the traditional SVM algorithm.
Table 6.5 Comparison Table for False Alarm Rate for SVM and GSVM-RU

<table>
<thead>
<tr>
<th>Attacks/Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>1.92</td>
<td>0.68</td>
<td>1.32</td>
<td>2.56</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>1.35</td>
<td>0.35</td>
<td>0.69</td>
<td>0.64</td>
</tr>
</tbody>
</table>

From the table 6.5, SVM for the four attacks of DoS, Probing, U2R and R2L gives higher alarm rate of 1.92, 0.68, 1.32 and 2.56. For the proposed GSVM-RU, it produces minimum rate of 1.35, 0.35, 0.69 and 0.64 which are better than SVM alarm rate.

![Figure 6.7 Comparison Graph for False Alarm Rate for SVM and GSVM-RU](image)

**Figure 6.7 Comparison Graph for False Alarm Rate for SVM and GSVM-RU**

From the figure 6.7, it can be seen that for the four attacks DoS, Probe, U2R and R2L the false alarm rate of GSVM-RU produces minimum value when compared with traditional SVM.
c) **Accuracy**

In this part the accuracy of the proposed method GSVM-RU is compared with the traditional SVM algorithm.

**Table 6.6 Comparison Table of Accuracy for SVM and GSVM-RU**

<table>
<thead>
<tr>
<th>Methods</th>
<th>DoS</th>
<th>Probe</th>
<th>U2R</th>
<th>R2L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>84</td>
<td>83.1</td>
<td>85</td>
<td>82</td>
</tr>
<tr>
<td>GSVM-RU</td>
<td>95</td>
<td>95.5</td>
<td>96</td>
<td>97</td>
</tr>
</tbody>
</table>

From the table 6.6, SVM for the four attacks of DoS, Probing, U2R and R2L gives accuracy of 84, 83.1, 85 and 82. For the proposed GSVM-RU, it gives 95, 95.5, 96 and 97 which are better than SVM accuracy.

![Comparison Graph of Accuracy for SVM and GSVM-RU](image)

**Figure 6.8 Comparison Graph of Accuracy for SVM and GSVM-RU**

From the figure 6.8, it can be seen that for the four attacks DoS, Probe, U2R and R2L accuracy of GSVM-RU produces higher accuracy when compared with traditional SVM.
6.5 SUMMARY

This chapter focuses on Intrusion Detection System using GVSM-RU classification of attacks. It mainly deals with the GSVM-RU classifier for the datasets. The experimental results of table and graph obtained for the proposed GSVM-RU technique for intrusion detection of attacks are also given along with the discussions.