Fast and Effective Spam Sender Detection with Granular SVM on Highly Imbalanced Mail Server Behavior Data

(Invited Paper)

Yuchun Tang Sven Krasser Paul Judge
Secure Computing Corporation
4800 North Point Parkway, Suite 400
Alpharetta, GA 30022
{ytang, skrasser, pjudge}@ciphertrust.com

Yan-Qing Zhang
Department of Computer Science
Georgia State University
Atlanta, GA 30302-3994
yzhang@cs.gsu.edu

***Abstract*—Unsolicited commercial or bulk emails or emails containing virus currently pose a great threat to the utility of email communications. A recent solution for filtering is reputation systems that can assign a value of trust to each IP address sending email messages. By analyzing the query patterns of each participating node, reputation systems can calculate a reputation score for each queried IP address and serve as a platform for global collaborative spam filtering for all participating nodes. In this research, we explore a behavioral classification approach based on spectral sender characteristics retrieved from such global messaging patterns. Due to the large amount of bad senders, this classification task has to cope with highly imbalanced data. In order to solve this challenging problem, a novel Granular Support Vector Machine – Boundary Alignment algorithm (GSVM-BA) is designed. GSVM-BA looks for the optimal decision boundary by repetitively removing positive support vectors from the training dataset and rebuilding another SVM. Compared to the original SVM algorithm with cost-sensitive learning, GSVM-BA demonstrates superior performance on spam IP detection, in terms of both effectiveness and efficiency.**

Keywords—spam filtering, data mining, class imbalance, granular support vector machine

I. INTRODUCTION

As spam volumes are continuing to increase, the need for fast and accurate systems to filter out malicious emails and allow good emails to pass through results in a great motivation for the development of email reputation systems. These systems assign a reputation value to each sender based on e.g. the IP address from which a message originated. This value reflects the trustworthiness of the sender. Traditional content filtering anti-spam systems can provide highly accurate detection rates but are usually prohibitively slow and poorly scalable to deploy in high-throughput enterprise and ISP environments. Reputation systems can provide more dynamic and predictive approaches to not only filtering out unwanted mails but also identifying good messages, thus reducing the overall false positive rate of the system. In addition, reputation systems provide real-time collaborative sharing of global intelligence about the latest email threats, providing instant protection benefits to the local analysis that can be performed by a filtering system.

One such reputation system is Secure Computing's TrustedSource™ [1]. TrustedSource operates on a wide range of data sources including proprietary and public data. The latter includes public information obtained from DNS records, WHOIS data, or real-time blacklists (RBLs). The proprietary data is gathered by over 5000 IronMail™ appliances that give a unique view into global enterprise email patterns. Based on the needs of the IronMail administrator, the appliance can share different levels of data and can query for reputation values for different aspects of an email message. Currently, TrustedSource can calculate a reputation value for the IP address a message originated from, for URLs in the message, or for message fingerprints. The classification algorithm presented in this research uses queries for IPs as input to detect spam sender IPs among them.

The rest of the paper is organized as follows. Section 2 describes the data we use to build the classifier. Section 3 gives a brief review for research on imbalanced classification. In Section 4, GSVM-BA is presented in detail. Section 5 reports performance numbers of GSVM-BA on IP classification. Finally, Section 6 concludes the paper.

II. DATA AND FEATURES CONSIDERED

A. Raw Data

For this research, we consider the simplest case in which the system is queried by appliances for IP addresses and the query information is fed back into the system for analysis. When an email message is received by an IronMail appliance, it will query the TrustedSource system for the reputation of this IP address. Hence, the query strongly correlates to a message transmission. This query type will give the system three pieces of information: the queried IP (Q), the source of the query (S), and the timestamp (T). Each QST tuple indicates that Q tried to send a message to S at time T.

The level of data in this type of query is akin to the level of data in a standard RBL IP lookup. Some previous work has

been done to detect malicious hosts based on the patterns of such queries by Ramachandran et al. in [2]. They investigate a dataset of RBL queries to spot exploited machines in botnets that are querying an RBL to test whether members of the same botnet have been blacklisted.

B. Feature Generation

Based on the QST data, we generate two sets of feature vectors. The first is called the breadth vector; the second is called the spectral vector.

The breadth vector contains information regarding to how many different IronMail appliances a particular IP tries to send, how many messages it sends, how many sending sessions (bursts) each IronMail saw, how many messages are in a burst, and how many global active periods an IP had. This data is based on the Q, S, and T information.

The spectral vector is concerned with the actual sending pattern of an IP. For this data, only Q and T are considered. For each IP, we look at a configurable timeframe t and divide it into N slices. This results in a sequence c_n where c_n is the number of messages received in slice n. This sequence is then transformed into the frequency domain using a discrete Fourier transform (DFT). Since we do not consider time zones or time shifts, we are only interested in the magnitude of the complex coefficient of the transformed sequence and throw away the phase information. This results in the sequence C_k as shown in (1).

$$C_k = \left|F\{c_n\}\right| = \left|\frac{1}{N}\sum_{r=0}^{N-1} c_r e^{-i2\pi krN^{-1}}\right| \qquad (1)$$

C_0 is the constant component that corresponds to the total message count, which is already part of the breadth data. We do not want to consider it for the spectral pattern again and normalize the remaining coefficients by it resulting in a new sequence $C'_k = C_k / C_0$. Furthermore, since the input sequence c_n is real, all output coefficients C_k with $k > N/2$ are redundant due to the symmetry properties of the DFT and can be ignored.

To build an example classifier, we chose t = 24h and N = 48. Each c_n holds then the message counts for an IP during a 30 minute time window. This results in 24 usable raw spectral features, C'_1 to C'_{24}.

An evaluation of these raw features indicated a lot of energy in the higher coefficients of spam senders. This means that spam senders do not have a regular low frequency sending behavior in our 24 hour time window. We achieved good results with derived features based on the raw factors that take this observation into account. We grouped C'_2 to C'_4 (Group 1) and C'_5 to C'_{24} (Group 2) and use the mean and standard deviation of both groups as four additional features per IP. C'_1 is considered separately. To evaluate these features, the Signal-to-Noise Ratio (S2N), defined as the distance of the arithmetic means of the spam and non-spam classes divided by the sum of the corresponding standard deviations [3], is used. Table I shows these values. Fig. 1 and Fig. 2 show the density of two of these derived features.

TABLE I. SIGNAL TO NOISE RATIO OF DERIVED FEATURES

Feature	S2N
Group 1 Mean	0.580
Group 1 Std Dev	0.232
Group 2 Mean	0.404
Group 2 Std Dev	0.448

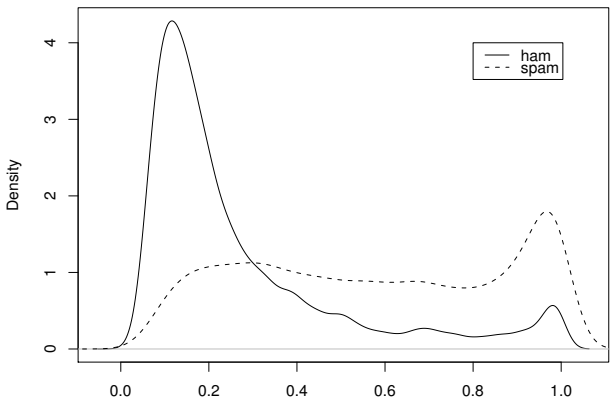

Figure 1. Density of the group 1 mean.

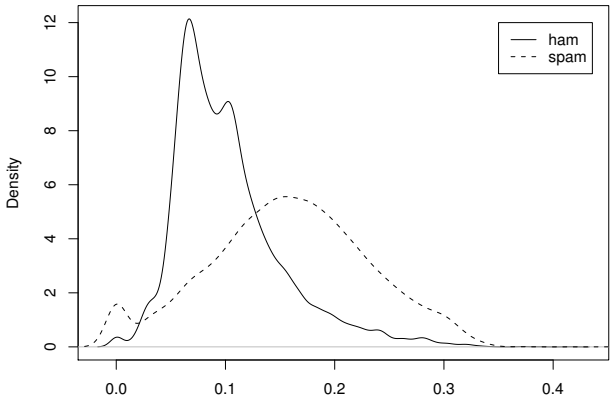

Figure 2. Density of the group 2 standard deviation.

Both S2N and the density analysis show that these derived features are potentially informative to discriminate spam IPs from non-spam ones.

III. IMBALANCED CLASSIFICATION

How to build an effective and efficient model on a huge and complex dataset is a major concern of the science of knowledge discovery and data mining. With emergence of new data mining application domains such as message security, e-business and biomedical informatics, more challenges are coming. Among them, highly skewed data distribution has been attracting noticeably increasing interest from the data mining community due to its ubiquitousness and importance [4-12].

TABLE II. QST DATA DISTRIBUTION AVERAGED ON 08/01/2006-08/31/2006

	Ratio
Spam IPs	95.57%
Non-spam IPs	4.43%

A. Class Imbalance

Class imbalance happens when the distribution on the available dataset is highly skewed. This means that there are significantly more samples from one class than samples from another class for a binary classification problem. Class imbalance is ubiquitous in data mining tasks, such as diagnosing rare medical diseases, credit card fraud detection, intrusion detection for national security, etc.

For spam filtering, each IP is classified as spam or non-spam based on its sending behavioral patterns. This classification is highly imbalanced. As shown in Table II, over 95% of the IPs are spam IPs. As our current target in this research is to detect spam IPs, we define spam IPs as positive and non-spam IPs as negative.

B. Metrics for Imbalanced Classification

Many metrics have been used for performance evaluation of classifications. All of them are based on the confusion matrix as shown at Fig. 3. With highly skewed data distribution, the overall accuracy metric defined by (2) is not sufficient any more. For example, a naive classifier that identifies all samples as spam will have a high accuracy of 95%. However, it is totally useless to detect spam IPs because all negative IPs are also identified as spam IPs by mistake (too many FPs).

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2}$$

To get optimal balanced classification ability, sensitivity as defined in (3) and specificity as defined in (4) are usually adopted to monitor classification performance on two classes separately. Notice that sensitivity is sometimes called true positive rate or positive class accuracy while specificity is called true negative rate or negative class accuracy. Based on these two metrics, g-mean has been proposed in [11], which is the geometric mean of classification accuracy on negative samples and classification accuracy on positive samples. Its definition is shown in (5). The Area Under ROC curve (AUC-ROC) metric [13] can also indicate a classifier's balance ability between sensitivity and specificity as a function of varying a classification threshold.

	predicted positives	predicted negatives
real positives	TP	FN
real negatives	FP	TN

Figure 3. The confusion matrix.

$$sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$specificity = \frac{TN}{TN + FP} \tag{4}$$

$$g - mean = \sqrt{sensitivity \times specificity} \tag{5}$$

Both g-mean and AUC-ROC can be used if the target is to optimize classification performance with balanced positive class accuracy and negative class accuracy.

On the other hand, sometimes we are interested in highly effective detection ability for only one class. For example, for credit card fraud detection problem, the target is detecting fraudulent transactions. For diagnosing a rare disease, what we are especially interested in is to find patients with this disease. For such problems, another pair of metrics, precision as defined in (6) and recall as defined (7), is often adopted. Notice that recall is the same as sensitivity. The f-value defined in (8) is used to integrate precision and recall into a single metric for convenience of modeling. Similarly, the Area Under Precision/Recall Curve (AUC-PR) [14] is also used to indicate the detection ability of a classifier between precision and recall as a function of varying a classification threshold.

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$f\text{–}value = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{8}$$

Because our target is to detect spam IPs, metrics concerning one class detection ability are more suitable than metrics concerning balanced classification ability. However, if the precision is too low, the classifier has no practical usefulness. For our application, it is required that the precision is higher than 99.8%. With this prerequisite in mind, we are also trying to increase the recall.

C. Methods for Imbalanced Classification

Many methods have been proposed for imbalanced classification and some good results have been reported [4]. These methods can be categorized into three different classes: cost sensitive learning, boundary alignment, and sampling. Sampling can be further categorized into two subclasses:

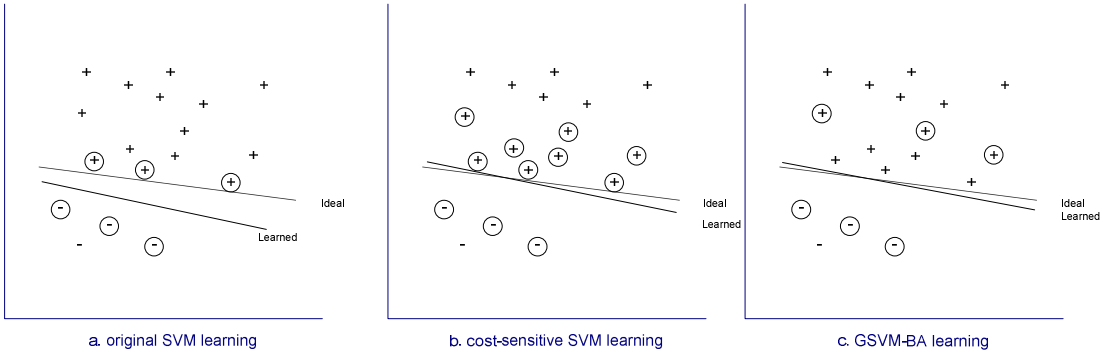

Figure 4. GSVM-BA can push the boundary back close to the ideal position with fewer SVs. The dotted line is the ideal boundary and the solid line is the learned boundary.

oversampling the minority class or undersampling the majority class. Interested readers may refer to [5] for a good survey.

For a real world classification task like spam IP detection, there are usually a large amount of IP samples. These samples need to be classified quickly so that spam messages from those IPs can be blocked in time. However, cost sensitive learning, boundary alignment, or oversampling usually increases modeling complexity and results in slow classification. On the other hand, undersampling is a promising method to improve classification efficiency. Unfortunately, random undersampling may not generate accurate classifiers because informative majority samples may be removed.

In this paper, granular computing and the support vector machine (SVM) algorithm are utilized for undersampling by keeping informative samples while eliminating irrelevant, redundant, or even noisy samples. After undersampling, data is cleaned and hence a good classifier can be modeled for IP classification both in terms of effectiveness and efficiency.

D. SVM for Imbalanced Classification

SVM embodies the Structural Risk Minimization (SRM) principle to minimize an upper bound on the expected risk [15,16]. Because structural risk is a reasonable trade-off between the training error and the modeling complication, SVM has a great generalization capability. Geometrically, the SVM modeling algorithm works by constructing a separating hyperplane with the maximal margin.

Compared with other standard classifiers, SVM performs better on moderately imbalanced data. The reason is that only support vectors (SVs) are used for classification and many majority samples far from the decision boundary can be removed without affecting classification [7]. However, performance of SVM is significantly deteriorated on highly imbalanced data. For this kind of data, the SRM principle is prone to find the simplest model that best fits the training dataset. Unfortunately, the simplest model is exactly the naive classifier that identifies all samples as majority.

Previous research that aims to improve the effectiveness for SVM includes the following:

Raskutti et al. explored effects of different imbalanced compensation techniques on SVM [8]. They demonstrated that a one-class SVM that learned only from the minority class sometimes can perform better than an SVM modeled from two classes.

Akbani et al. proposed the SMOTE with Different Costs algorithm (SDC) [7]. SDC conducts oversampling on the minority class by applying Synthetic Minority Oversampling TEchnique (SMOTE) [10], a popular oversampling algorithm, with different error costs. As a result, the boundary of the learned SVM can be better defined and far away from the minority class.

Wu et al. proposed the Kernel Boundary Alignment algorithm (KBA) that adjusts the boundary toward the majority class by directly modifying the kernel matrix [9]. In this way, the boundary is expected to be closer to the "ideal" boundary.

The problem of the one-class SVM is that it actually performs worse in many cases compared to a two-class SVM. SDC and KBA are also not suitable for IP classification because they usually take a longer time for classification than SVM. As demonstrated in our empirical studies, SVM itself is already too slow to be adopted for IP classification.

The speed of SVM classification depends on the number of SVs. For a new sample x, $K(x,sv)$ is calculated for each SV. Then it is classified by aggregating these kernel values with a bias. To speed up SVM classification, one potential method is to decrease the number of SVs.

IV. GSVM-BA ALGORITHM

In this work, a novel Granular Support Vector Machines-Boundary Alignment algorithm (GSVM-BA) is designed with the principle of granular computing to build a more effective and more efficient SVM for IP classification.

A. Granular Computing and GSVM

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 [17,18]. There are two principles in granular computing. 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"). As opposed to traditional data-oriented numeric computing, granular computing is knowledge-oriented [18]. By embedding prior knowledge or prior assumptions into the granulation process for data modeling, better classification can be achieved.

A granular computing-based learning framework called Granular Support Vector Machines (GSVM) was proposed in [19]. GSVM combines the principles from statistical learning theory and granular computing theory in a systematic and formal way. GSVM works by extracting a sequence of information granules with granule split and/or granule shrink, and then building an SVM on some of these granules when necessary.

The main potential advantages of GSVM are:

- Compared to SVM, GSVM is more adaptive to the inherent data distribution by trading off between local significance of a subset of data and global correlation among different subsets of data, or trading off between information loss and data cleaning. Hence, GSVM may improve the classification performance.
- GSVM may speed up the modeling process and the classification process by eliminating redundant data locally. As a result, it is more efficient on huge datasets.

B. *GSVM-BA*

The data cleaning principle of granular computing makes GSVM-BA ideal for spam IP detection on highly imbalanced data derived from QST data.

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 toward the minority class [7,9]. As demonstrated in Fig. 4(a), positive SVs that are close to the learned boundary may be noisy. Some very informative samples may hide behind them.

To find these informative samples, we can conduct cost-sensitive learning to assign more penalty values to false positives (FP) than false negatives (FN). This method is named SVM-CS. However, SVM-CS increases the number of SVs (Fig. 4(b)), and hence slows down the classification process.

In contrast to this, GSVM-BA looks for these informative samples by repetitively removing positive support vectors from the training dataset and rebuilding another SVM. GSVM-BA embeds the "boundary push" assumption into the modeling process. After an SVM is modeled, a "granule shrink" operation is executed to remove corresponding positive SVs to produce a smaller training dataset, on which a new SVM is modeled. This process is repeated to gradually push the boundary back to its ideal location, where the optimal classification performance is achieved.

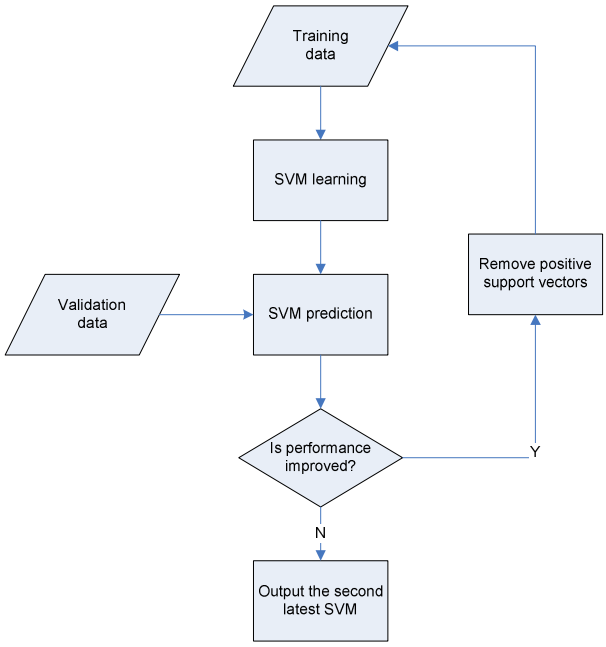

Figure 5. GSVM-BA algorithm.

Empirical studies in the next section show that GSVM-BA can compute a better decision boundary with much fewer samples involved in the classification process (Fig. 4(c)). Consequently, classification performance can be improved in terms of both effectiveness and efficiency.

Fig. 5 sketches the GSVM-BA algorithm. The first SVM is always the naive one by default.

V. EXPERIMENTS

For spam IP detection experiments, we run SVM-CS and GSVM-BA on a workstation with a Pentium® M CPU at 1.73 GHz and 1 GB of memory.

A. *Data Modeling*

The QST data gathered on 07/31/2006 is used for training and the QST data from between 07/24/2006 and 07/30/2006 is used for validation. The training dataset is normalized so that the value of each input feature is falls into the interval of *[-1,1]*. The validation dataset is normalized correspondingly. An SVM with the RBF kernel is adopted with parameters (γ, C) optimized by grid search [20] on the validation dataset.

For SVM-CS, the cost for the positive class is always 1 and the cost for the negative class is tuned and optimized on the validation dataset. Similarly, the number of times the "granular shrink" operation is executed is also optimized on the validation dataset for GSVM-BA.

After an SVM is modeled, it is applied on the normalized QST data gathered between 08/01/2006 and 08/31/2006 and the corresponding performance is reported.

TABLE III. EFFECTIVENESS COMPARISON ON 08/01/2006-08/31/2006

	SVM-CS	GSVM-BA
Precision	99.81±0.07%	99.87±0.06%
Recall	42.02±5.96%	47.14±5.97%

TABLE IV. EFFICIENCY COMPARISON ON 08/01/2006-08/31/2006

	SVM-CS	GSVM-BA
#SVs	10393	581
Classification time	>3 hours	10 minutes

B. Result Analysis

Table III shows the effectiveness of GSVM-BA and SVM-CS. GSVM-BA is better in terms of both precision and recall. GSVM-BA can identify about 129K TPs and 160 FPs while SVM-CS catches about 115K TPs and 220 FPs everyday. Note that many FPs reported by GSVM-BA are sending both spam and legitimate messages. Hence, these IPs are actually spam senders, but since they send ham messages as well, we do not want to catch them with this classifier.

As demonstrated in Table IV, GSVM-BA takes about 10 minutes for classification with only 581 SVs while SVM-CS needs more than 3 hours with more than 10K SVs. The superior efficiency of GSVM-BA is crucial for our application to detect and block spam senders on time before they can send large amount of messages to TrustedSource-enabled mail servers.

VI. CONCLUSIONS

A new GSVM modeling algorithm, named GSVM-BA, is designed specifically to deal with highly imbalanced QST data based on global messaging patterns. After extracting informative breadth and spectral features from simple QST sending patterns, GSVM-BA works to gradually push the decision boundary back to its "ideal" position. In this modeling process, the dataset is cleaned and hence fewer samples/support vectors are involved in the classification process thereafter. Compared to SVM with cost-sensitive learning, GSVM-BA is both efficient and effective as our empirical studies demonstrated. The resulting classifier contributes to the TrustedSource reputation system as one source of input and prevents malicious or unwanted messages being delivered to email users.

REFERENCES

[1] Secure Computing TrustedSource Website, http://www.trustedsource.org.

[2] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using DNSBL counter-intelligence," Proc. 2nd USENIX Steps to Reducing Unwanted Traffic on the Internet, 2006, pp. 49-54.

[3] T. Furey, N. Cristianini, N.Duffy, D. Bednarski, M. Schummer, and D. Haussler, "Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data," Bioinformatics, Vol. 16, 2000, pp. 906-914.

[4] N. V. Chawla, N. Japkowicz and A. Kolcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," SIGKDD Explorations, Vol. 6, No. 1, 2004, pp. 1-6.

[5] G. M. Weiss, "Mining with Rarity: A Unifying Framework," SIGKDD Explorations, Vol. 6, No. 1, 2004, pp. 7-19.

[6] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," Intelligent Data Analysis, Vol. 6, No. 5, 2002, pp.429-450.

[7] R. Akbani, S. Kwek, and N. Japkowicz, "Applying Support Vector Machines to Imbalanced Datasets," Proc. of the 2004 European Conference on Machine Learning (ECML2004), 2004, pp.39-50.

[8] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for SVMs: a case study," SIGKDD Explorations, Vol. 6, No. 1, 2004, pp. 60-69.

[9] G. Wu and E. Y. Chang, "KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, 2005, pp. 786-795.

[10] N.V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling TEchnique," Journal of Artificial Intelligence Research, Vol. 16, 2002, pp. 321-357.

[11] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One sided selection," Proc. of the 14th International Conference on Machine Learning (ICML1997), 1997, pp.179-186.

[12] N. V. Chawla, A. Lazarevic, L. O. Hall and K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," Proc. of Principles of Knowledge Discovery in Databases, 2003, pp. 107-119.

[13] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," Pattern Recognition, Vol. 30, No. 7, 1997, pp. 1145-1159.

[14] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," Proc. of the 23rd International Conference on Machine Learning (ICML2006), 2006, pp. 233-240.

[15] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.

[16] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, Vol. 2, No. 2, 1998, pp. 121-167.

[17] A. Bargiela, Granular Computing: An Introduction, Kluwer Academic Pub, Kluwer, 2002.

[18] J. T. Yao, Y. Y. Yao, "A granular computing approach to machine learning," Proceedings of FSKD'02, Singapore, 2002, pp. 732-736.

[19] Y.C. Tang, B. Jin and Y.-Q. Zhang, "Granular Support Vector Machines with Association Rules Mining for Protein Homology Prediction," Special Issue on Computational Intelligence Techniques in Bioinformatics, Artificial Intelligence in Medicine, Vol. 35, No. 1-2, 2005, pp. 121-134.

[20] C.-W. Hsu, C.-C. Chang, C.-J. Lin, "A practical guide to support vector classification," Available at http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.