Effects of Distance between Classes and Training Datasets Size to the Performance of XCS: Case of Imbalance Datasets

Thach H. Nguyen, Sombut Foitong, Sornchai Udomthanapong and Ouen Pinngern

***Abstract*—** This paper analyzes the effects of distance between classes and training datasets size to XCS classifier system on imbalanced datasets. Our purpose is to answer the question whether the loss of performance incurred by the classifier faced with class imbalance problems stems from the class imbalance per se or it can be explained in some other ways. The experiments from 250 artificial imbalanced datasets show that XCS can perform well in some imbalance domains if the training datasets size is large enough and the distance between classes is appropriate. Thus, it dose not seem fair to correlate imbalance datasets directly to the loss performance of XCS. Through this research, we also know what kinds of datasets are suitable for training XCS and dealing with class imbalances alone will not always help improve performance of classifiers.

***Keywords* -** Data mining, XCS, imbalance dataset, distance between classes.

I. Introduction

XCS is an accuracy-based learning classifier system (LCS) that is shown to perform very competitively with respect to other machine learning methods in classification problems. Different from other learning classifier systems executing genetic algorithm panmictically, XCS executes a genetic algorithm in niches defined by the action sets [3] that produces a more complete model of the problem space.

XCS has already attracted much attention from machine learning and data mining communities [1], [5]. There are some performance comparisons of strength-based fitness systems and accuracy-based fitness systems showing that accuracy-based fitness systems, including XCS, have a significant impact on data mining field. In [6], Mansilla and Garrel-Guiu studied the performance in a set of well-known machine learning problems demonstrating that XCS is able to produce a classification performance and rule set which exceeds the performance of most current Machine Learning techniques.

Recently, the class imbalance problem has been recognized as a crucial problem in machine learning and data mining. This problem has been reported to hinder the XCS's performance on many types of problems [6][7][8]. However, the main problem with these proposals is that they ignored the effects of training datasets, distance between classes and no study has made a point of linking the class imbalance problem directly to this loss. As a matter of fact, although the performance of XCS may decrease on many imbalance domains, that does not prove that it is imbalance domains per se, that causes that decrease. Rather it is quite possible that class imbalances yield certain conditions that hamper classification, which would suggest that 1) class imbalances are not necessarily always a problem and, perhaps even more importantly, 2) dealing with class imbalances will not always help improve performance of classifiers. Our experiments show that the distance between classes affects extremely to accuracy of learning systems on high class imbalance levels. And in this research, we also analyze the effects of training dataset size and determine how it affects to classifier learning system, XCS.

The purpose of this paper is to answer the question whether class imbalances are truly to blame for the reported losses of XCS performance or whether these deficiencies can be explained in some other ways. We show that class imbalances are, actually, not a problem by themselves, but that, in small and complex datasets, they come accompanied with the problem of distance between classes which in turn causes degradation in XCS's performance.

The remainder of this paper is organized as follows. The application of data mining techniques to class imbalance problems are discussed in section 2 which is followed by a brief description of XCS. From section 4, dataset generation with all parameters: imbalance rate – *ir*, distance between classes – *dis* and training dataset size – *s* will be described. In section 5, the results and experiments of XCS in different dataset domains will be presented. Section 6 concludes this work and directs the future works.

II. Related Work

To solve class imbalance problems, many methods and

Manuscript received October 15, 2007. This work was sponsored by the AUN/SEED-Net organization and the Information Science Laboratory, Research Center for Communications and Information Technology: ReCCIT, King Mongkut's Institute of Technology Ladkrabang, Thailand.

T. H. Nguyen, is a master student in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand (e-mail: nguyenhuythach1983@yahoo.com).

S. Foitong is a doctoral student in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand (e-mail: sombutf@yahoo.com).

S. Udomthanapong is a master student in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand (e-mail: u.sornchai@gmail.com).

O. Pinngern is an Associate Professor in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand (e-mail: kpouen@kmitl.ac.th).

algorithms have been proposed. Some of the best well-known approaches are applied at the sampling level. In [10], Chawla et al. presented SMOTE which can be done either over-sampling minority class or under-sampling the majority class. Both methods can be applied in any concept learning system, since they act as a preprocessing phase, allowing the learning system to receive the training instances as if they belonged to a well-balance dataset. Thus, any bias of the system towards the majority class due to the different proportion of examples per class would be expected to be suppressed. In addition, in [12], the effects of sampling method, probabilistic estimate and decision tree structure of C4.5 on imbalance datasets were investigated. They also proposed an average improvement of the Area under the ROC curve (AUC) measure over the other sampling schemes. Beside that, G.M. Weiss and F. Provost [16] analysis the effects of imbalance datasets to classifier learning systems and how they affect through the evaluation of learned classifiers by using two performance measures: AUC and classification accuracy. In [13], a new approach in cost-sensitive to neural networks is proposed instead of decision trees to solve multiclass tasks. And in [17], G.M. Weiss indicated the learning from imbalance and rarity datasets can be handled in a similar manner. Recently, Chris S. et al. [9] evaluated the performance of seven commonly-used sampling techniques on imbalance and noise datasets to study the effects of noise to imbalance problem. The most notice researches are [14] and [15]. They proposed a new sight in imbalance problem by proposing to consider the effects of overlapping datasets and small disjunction problem. This approach is also focused in our research.

In learning classifier systems field, there are 3 different approaches: Fitness adaptation, population size and parameter settings. In [8], Albert O.P. and Ester B.M showed that XCS with standard parameter settings for 10 imbalance levels, the True Negative (TN or majority class) rate quickly reaches 100%, but the True Positive (TP or minority class) rate raises to 100% for imbalance levels up to i=4 (i is the imbalance level used to calculate the imbalance ratio $ir=2^i$). For i=5, XCS needs a long training time to reach 100% correct of TP rate and, for $i \geq 6$, the system classifies all input instances belonged to the majority. At that time, the population mainly consists of the two *overgeneral* rules: ######:0 and ######:1 (in case of 6-multiplexer).

To overcome this problem, tuning XCS's parameters based on the dataset imbalance ratio, is proposed by 2 methods: manual (offline) and automatic (online) adjustment. The experiments show that XCS can solve the unbalanced 11-bit multiplexer problem until i=8, which is a notable improvement with respect to the initial experiments.

Another attribution deals with the class imbalance problem by finding fitness adaption [7] based on class-sensitive accuracy in combination with UCS, a supervised LCS derived from XCS, as a useful tool for alleviating the effects of class imbalances. In fact, because XCS, UCS and some other learning classifier systems have their fitness based on accuracy, they present a high bias toward the majority class instances and evolve easily *overgeneral* classifiers. Idea of the proposal is restrict classifiers to cover regions formed by examples of a single class and make accuracy class-sensitive rather than instance-sensitive. Thus, accuracy is modified so that each class is considered equally important regardless of the number of instances representing in each class. The experiments show that this model evolved with imbalance level up to i=7.

III. XCS Overview

Like other classifier systems, XCS seeks a reinforcement reward from its environment based on an evolving set of condition-action rules called classifiers. Via genetic algorithm process, classifiers useful in gaining reinforcement are selected and propagate over those less useful leading to increasing system performance. Figure 1 illustrates a broad picture of XCS.

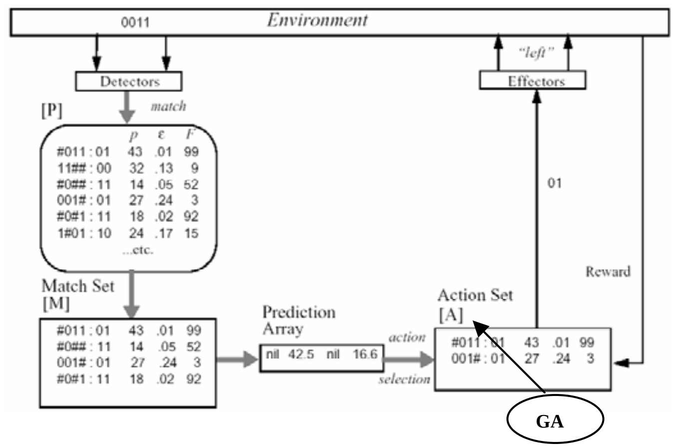


Figure 1: Schematic illustration of XCS in classification application.

In the training time, datum is read from the training data set and encoded to a classifier by detectors and it is matched with classifiers of population [P] in the condition part to form match set [M]. From the match set, the prediction value P(a_i) for each action a_i appearing in [M] is calculated. The P(a_i) values are placed in a prediction array and an action is selected by some methods as: selecting action with the largest prediction is called deterministic action selection or exploit method, roulette-wheel action selection or selecting completely random. Once an action is selected, the action set [A] is formed. The action in [A] will be sent to environment by the effectors and a reward r be returned by the environment. The Q-learning method of Reinforcement learning and MAM technique use the reward r to update values of classifiers in the action set [A]. Because classification problem is a single step problem, Genetic Algorithm (GA) is applied in [A] instead of $[A]_{-1}$, previous action set, as multi-step problems [3]. The more details are referred to [3] and a detailed algorithmic description can be found in [4].

IV. DATASET GENERATION

Because our purpose is to understand how class imbalances influence to the performance of XCS, we chose to run our experiments on a series of artificial domains whose characteristics could be controlled carefully rather than conducting our study on real-world domains whose results would be difficult to decipher.

The artificial datasets employed in the experiments have three major controlled parameters. The first one is distance between means of two classes in Gaussian distribution (*dis*), the second one is the training dataset size level (*s*) and the last one is the grade of imbalance (*i*). In particular, we created 250 domains belong to change of the three parameters by employing the method represented in [14]. In [14], Nathalie Japkowics designed a similar framework for testing the effect of the small disjunction to the imbalance dataset problem. In addition, in [15], R.C. Prati et al. studied the relationship between the problem of overlapping the data and dealing with class imbalances with the same method. It seemed reasonable to assume that this framework would apply to our case as well.

The 250 generated domains of our study were generated in the following way: each of the domains is two-dimensional (two attributes), and each attribute value is generated at random, using a Gaussian distribution, with standard deviation equal to value 1. Jointly, each domain has two classes: positive and negative. The mean of the positive class is created firstly and based on the distance between classes; the mean of the negative class is found. After that, belong to the value of training dataset size level *s* and imbalance level *i*, we calculate the number of instances in majority class (n_{maj}) and minority class (n_{min}). Actual training datasets are generated from this backbone model.

Ten differences of distance between classes were considered (*dis=0..9*). For the first domain, the means of the Gaussian function for both classes are similar. For the following domains, we stepwise add 1 standard deviation to the mean of the negative class, up to 9 standard deviations.

Five training dataset sizes were considered (*s=1..5*) where each size, s, corresponds to the number of instances in the majority class: $round((8192/2^5)*2^s)$. We will explain why we chose the number 8192 later.

Finally, five levels of class imbalance were also considered. Because, with standard parameter settings, XCS can solve well problems with imbalance level $i \leq 4$ as introduced in the previous section, in this paper, the imbalance level *i* is considered only with *i=5..9*. Each level *i* corresponds to the situation that minority class contains only $1/2^i$ of all instances in the majority class. So with given values of *s* and *i*, we can calculate the number of instances in majority and minority are $n_{maj}=round((8192/2^5)*2^s)$ and $n_{min}=round((8192/2^5)*2^s/2^i)$, respectively. For example, for *s=1* and *i=5*, the majority class is represented by $n_{maj}=round((8192/2^5)*2^1)=512$ instances and the minority class is represented by $n_{min}=round((8192/2^5)*2^1/2^5)=16$ instances; If *s=5, i=9* then $n_{maj}=8192$ and $n_{min}=16$.

Now, we explain why we chose the number 8192 in formulation $n_{maj}=round((8192/2^5)*2^s)$. In the initial time, we did not know that number, we called it is *x*, and we had to find it. By this framework, we have number of instances in minority class depending on *x*: $n_{min}=round((x/2^5)*2^s/2^i)$. To ensure XCS can learn the condition $n_{min} \geq 1$ must be approved. We see that, in the most extremely imbalance level case and the smallest training dataset size: *s=1, i=9,* $n_{min} = round((x/2^5)*2^s/2^i) = round((x/2^5)*2^1/2^9) = round(x*2^{-13}) \geq 1$, or $x \geq 8912$. In our research, we chose *x*=8912.

V. EXPERIMENTS AND RESULTS

To answer the question whether the class imbalance problem always causes degradation in performance of XCS or it does so only in certain cases, we ran XCS [3], [4] on the artificial datasets described in the previous section and XCS's parameters settings as table 1. These values are described in [4]:

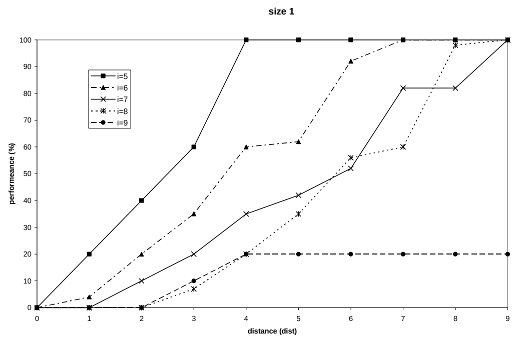
TABLE I: XCS'S PARAMETERS SETTINGS

Symbol	Parameter	Value
N	Maximum size of population	2000
χ	Crossover probability in GA	0.75
μ	Mutation probability in GA	0.01
β	Learning rate	0.2
γ	Discount factor	0.95
θ_{GA}	GA threshold	25
$P_\#$	Covering probability of using # when covering	0.66

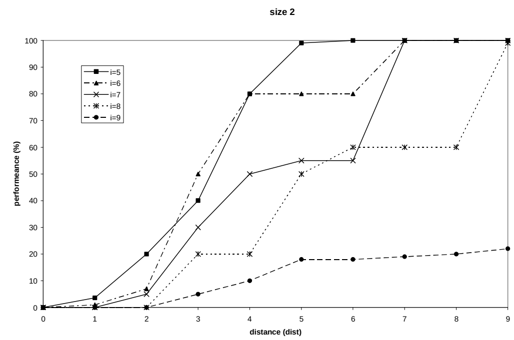
A. The effects of imbalance datasets to XCS

The results of our experiments are displayed in Figure 2 which plots the performance of XCS obtained for each combination of training dataset size, imbalance level and distance between classes. Each plot of this figure represents XCS performance obtained at a difference training set size. The top-leftmost plot corresponds to the smallest size (*s*=1) and progress until the below-center plot which corresponds to the largest size (*s*=5). Within each of these plots, each line connected by 10 points represents the concept of imbalance level from *i*=5 to *i*=9. Within each line, finally, each point corresponds to a particular distance between classes. The leftmost (or starting) point of each line corresponds to the nearest distance between two classes (*dis*=0) and progress until the rightmost point which corresponds to the farthest distance between classes (*dis*=9). The height of each point represents the average percent performance obtained by XCS on the training dataset size, imbalance level and distance between classes that this point represents.

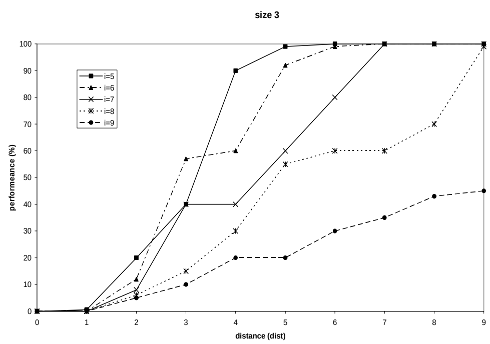
Our results reveal several points of interest: first, no matter what size of the training set is, linearly separable domains with domains of distance level *dis*=9 do not appear sensitive to any amount of imbalance (except *i*=9, we will analysis this case in the next section). As a matter of fact, as degree of



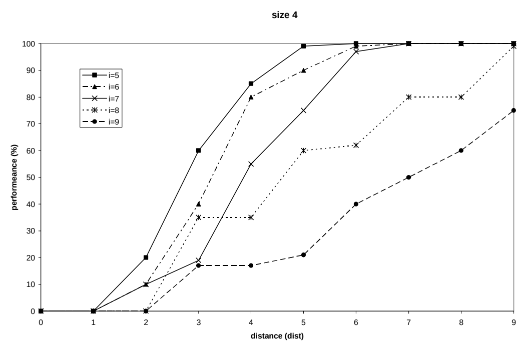
(a) Size 1



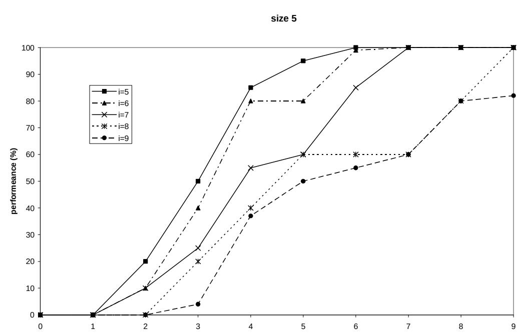
(b) Size 2



(c) Size 3

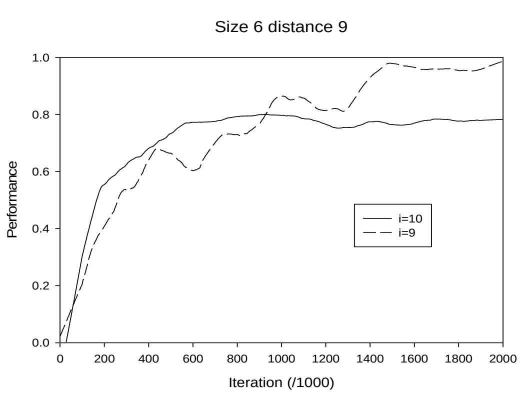


(d) Size 4

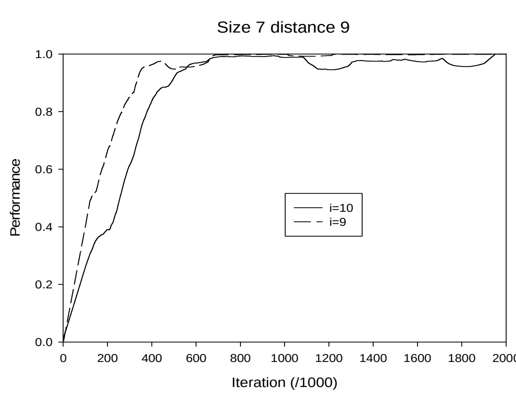


(e) Size 5

Figure 2: XCS and the Class Imbalance Problem



(a) Size 6



(b) Size 7

Figure 3: XCS and the training dataset size

distance between classes decreases, so does the XCS's sensitivity to imbalances. Indeed, we can clearly see in the Figure 2 that as the degree of distance decreases, XCS's performances are caused lower with higher of imbalance level.

As could be expected, imbalance rates are also a factor in the performance of XCS and, perhaps more surprisingly, so are the training set sizes. Indeed, as the size of the training set increases, the effects of imbalance decreases. This suggests that in very large domains, the class imbalance problem may not be a hindrance to a classification system. Specifically, the issue of relative cardinality of the two classes – which is often assumed to be the problem underlying domains with class imbalanced – may in fact be easily overridden by the use of a large enough dataset (if, of course, such a dataset is available and its size does not prevent the classifier from learning the domain in an acceptable time frame). This question is considered in more detail in the next section.

From our research, we can see that the imbalance problem may not always be to blame for the often observed XCS performance loss that accompanies it. Rather, we suggest that this performance loss may be cause by the small distance between classes with a small training dataset size. In other words, a huge class imbalance will not hinder classification of a domain whose distance between classes is far nor will we see a problem if the training dataset is very large. Conversely, a small dataset imbalance can greatly harm a very small dataset or one representing a very small distance between classes.

B. Class imbalance versus Training dataset size and Distance between classes

The experiments of this section were designed to verify the hypothesis proposed in the previous section and answer the question whether or not the loss of performance experienced by XCS is caused by the training dataset size and distance between classes rather than the class imbalance. In order to test this hypothesis in our artificial domains, we kept the same generation scheme with respect to concept of distance between classes, class imbalance and emphasized on the training set size. The amount of training data is known to affect on the performance of classifier systems. Providing additional training data typically leads to a more complex model (e.g. more rules, time training etc.) but with improving classification performance.

For a better visualization, we look up these results shown graphically in Figure 2, it shows that the training dataset size affects slightly to dataset containing low imbalance rate (i.e. *i*=5, 6) but significantly to dataset containing high imbalance rate (*i*=8, 9). This phenomenon is explained that in low imbalance rate datasets, the number of instances in minority class is enough for XCS learning. For example, if $i=5$, $s=1$, the number of instances in minority class is $n_{min} = round(2^{13}*2^{s}/(2^{5}*2^{i})) = round(2^{13}*2^{1}/(2^{5}*2^{5}))=16$ comparing to number of instances in majority class is $n_{maj} = round(2^{13}*2^{s}/2^{5}) = round(2^{13}*2^{1}/2^{5}) = 512$. However, in high imbalance rate datasets, the number of instances in minority class is very small and XCS encounters obstacle during training time so the performance is low. For example, if $i=9$, $s=1$; $n_{min}= round(2^{13}*2^{s}/(2^{5}*2^{i})) = round(2^{13}*2^{1}/(2^{5}*2^{9})) = 1$ comparing to $n_{maj} = round(2^{13}*2^{s}/2^{5}) = round(2^{13}*2^{1}/2^{5}) = 512$, or the imbalance ratio ir=1:512.

With extremely imbalance datasets: *i*=9 when *dis*=9, we increase the training dataset size from 513 instances (the top-left graph), correlative to *s*=1, to 8208 instances (the below-center), correlative to *s*=5, the performance of XCS increases from 20% to 82%.

In order to emphasis more detail on the function of training dataset size, we test performance of XCS in imbalance levels with respect to distance between classes, *dis* =9 and *s*=6 or *s*=7. In this case, we have the number of instances in minority class: $n_{min}= round(2^{13}*2^{s}/(2^{5}*2^{i})) = round(2^{13}*2^{6}/(2^{5}*2^{9})) = 32$ or $n_{min}= round(2^{13}*2^{s}/(2^{5}*2^{i})) = round(2^{13}*2^{7}/(2^{5}*2^{9})) = 64$ and the number of instances in the majority class $n_{maj}= round(2^{13}*2^{s}/2^{5}) = round(2^{13}*2^{6}/2^{5}) = 16384$ or $n_{maj}= round(2^{13}*2^{s}/2^{5}) = round(2^{13}*2^{7}/2^{5}) = 32768$ to implement the imbalance levels *i*=9 and *i*=10, respectively. We chose these values of *i* because the two imbalance levels represent two extremely imbalance cases in which we believe that many learners will suffer from biases to the majority class and according to [8], with standard or adjustment parameter settings, XCS encounters problem to solve these imbalance rates. However, by experiments showing in Figure 3, XCS can perform well in these cases. In all runs, XCS was trained during 4,000,000 learning iterations, but only the first 2,000,000 are shown for a better visibility. These results once again agree with our assertion. In training size level *s*=6 with imbalance level *i*=10, XCS reaches to 80% and with *i*=9, XCS reaches to 95% after $2*10^{6}$ iterations, stabilizing at 100% after $4*10^{6}$ explore trials. We increase the training size level to *s*=7, XCS even reaches to 100% performance with *i*=9 only after $6*10^{5}$ iterations and with *i*=10, XCS nearly reaches 100% after $2*10^{6}$ explore trials.

The goal of our experiments in the previous section and this section is to show that the performance of XCS, through hindered by class imbalance, is repaired as the training dataset size and distance between classes increase.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

In this work, we developed a systematic study using a set of artificially generated datasets with aim to answer question whether the loss of XCS's performance with imbalance problems is from the class imbalance itself or it is caused by other reasons. Our experiments show that the class imbalance by itself does not seem to constitute a crucial problem for XCS's performance. In fact, in the presence of imbalance with far distance between classes and large training dataset size, XCS provides high performance on both classes even in extremely imbalance datasets. In contrast, the combination of class imbalance, small distance between classes and small training dataset size make a degradation of XCS's classification accuracy.

Our future works are to extend the study to a more general class of problems, effects of injecting various degrees and types of noises to XCS's performance. Another important consideration has to do that is studying sampling techniques [10] in conjunction with XCS to enhance quality classification performance. Least but not last, it is imperative that we conduct experiments on a number of real-world datasets to verify that the hypothesis we posited on simple artificial datasets actually does apply to general and actual data.

ACKNOWLEDGMENT

We would like to thank to AUN-SEED/Net about the financial support for doing this research. We are also grateful to Research Center for Communications and Information Technology (ReCCIT) and Department of Computer Engineering, KMITL Ladkrabang for the help in research equipments.

REFERENCES:

[1] P.L. Lanzi, W. Stolzmann, S.W. Wilson, *Learning Classifier Systems: From Foundations to Applications*, Springer Berlin/Heidelberg, 2000.

[2] P.N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining,* Addison Wesley, USA, 2006.

[3] Stewart W. Wilson, "Classifier Fitness Based on Accuracy," *Evolutionary Computation*, Vol.3 (2), (1995), pp.149-175.

[4] Butz, M. and Wilson, S., "An algorithmic description of XCS", *Journal of Soft Computing 6*, (2002), pp.144-153.

[5] E. Bernadó-Mansilla, J.M. Garrell Guiu. "Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks". *Evolutionary Computation 11*, vol. 3, pp. 209-238, 2003.

[6] Orriols-Puig A., Goldberg D.E., Sastry K., and Bernadó-Mansilla E. "Modeling XCS in Class Imbalances: Population Size and Parameter Settings". *Technical report, IlliGAl Report* No. 2007001, USA, Feb. 2007.

[7] Albert Orriols and Ester Bernado-Mansilla, "Data Imbalance in UCS Classifier System: Fitness Adaptation", *IEEE Congress on Evolutionary Computation - CEC2005,* vol. 1, pp. 604-611, 2005.

[8] Orriols-Puig A. and Ester Bernadó-Mansilla, "Bounding XCS's Parameters for Unbalanced Datasets". *GECCO'06,* pp. 1561 - 1568. July 8–12, 2006, Seattle, Washington, USA.

[9] Chris S., Taghi M.K., Jason V.H., Andres F., "An Empirical Study of the Classification Performance of Learners on Imbalanced and Noisy Software Quality Data", *Information Reuse and Integration, 2007, IEEE International Conference,* pp. 651-658, August 2007.

[10] Nitesh V.C., Kevin W.B., Lawrence O.H., W. Philip K., "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research* 16 pp. 321–357, 2002.

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

[12] N. V. Chawla. "C4.5 and imbalanced datasets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure". *In Proceedings of the ICML'03 Workshop on Class Imbalances,* 2003.

[13] Zhi-Hua Zhou and Xu-Ying Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem", *Knowledge and Data Engineering, IEEE Transactions,* vol. 18(1), pp. 63-77, January 2006.

[14] Jo, T. and Japkowicz, N., "Class Imbalances versus Small Disjuncts", *SIGKDD Explorations* 6(1), June 2004.

[15] R.C. Prati, G.E.A.P.A. Batista, and M.C. Monard, "Class Imbalances *versus* Class Overlapping: An Analysis of a Learning System Behavior", *the Third Mexican International Conference on Artificial Intelligence - MICAI 2004*, LNAI 2972, pp. 312–321, 2004.

[16] Weiss, G. & Provost, F. "The Effect of Class Distribution on Classifier Learning: An Empirical Study", *Technical Report ML-TR-44, Department of Computer Science,* Rutgers University, 2001.

[17] G.M. Weiss, "Mining with Rarity - Problems and Solutions: A Unifying Framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7-19, 2004.

Thach Huy Nguyen received B.S. degree in information technology from Hanoi University of Technology, Vietnam, in 2006. Currently, he is a Master student under scholarship of AUN/SEED-Net, JICA, Japan, in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok 10520 THAILAND. His research interests include Learning Classifier System, Genetic Algorithm, Rough Set and Classification.

Sombut Foitong received the B.S. degrees in mathematical – computer from Ramkhamhaeng University and the M.Eng degrees in computer engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, THAILAND. Currently, he is a PhD student in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, THAILAND. His research interests include fuzzy classification, rule-based generation, rough set and soft computing.

Sornchai Udomthanapong received the B.S. degree in conputer engineering from King Mongkut's Institute of Technology Ladkrabang, Bankok, Thailand. Currently, he is a master degree student in the Department of Computer Engineering from King Mongkut's Institute of Technology Ladkrabang, Bankok 10520 Thailand. His research interests include classification Learning Classifier System, Genetic Algorithm, Neural Network, Data Mining.

Ouen Pinngern received the MS degree from Oregon State University in USA in 1983 and the PhD degree from University of Nebraska at Lincoln in USA in 1986 both in computer science. He is an associated professor in the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, THAILAND. His current research interests lie in the areas of soft computing, machine learning, data mining, information retrieval, and model of computation. He has been appointed as a Director of Research Center for Communications and Information Technology (ReCCIT) which is promoted to be a Center of Excellence in ICT under KMITL.