

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Artificial Endocrine System and Its Application for Nearest Neighbor Rule Condensation

Zhanmin Wang, Duwu Cui and Qingzheng Xu  
Xian University of Technology, China

---

**Abstract:** Prototype reduction algorithms are very useful in the area of nearest neighbor classification and most of them process the given datasets in its entirety. The main disadvantage of them is the excessive computational cost especially when the prototype size is very large. A divide-reduce-coalesce mechanism has been proposed to overcome this shortcoming. But, the final subsets obtained with most of the traditional algorithms in this divide-reduce-coalesce manner are not consistent subsets of original sample set. So, the accuracies of the classifiers obtained in the manner are not been guaranteed too. In this study, we present an artificial endocrine system to reduce useless points in a given training set. The method remains only the points on the boundary between different classes and the amount of reduced rules of the reference set can be revised by the granularity of the lattice. The main advantages of the proposed method are, on the one hand that it is able to condense a given rule set within less time comparing with other traditional algorithms. On the other hand, with the proposed algorithm we can get a consistent subset of the given set in the divide-reduce-coalesce manner. This approach has been tested using 13 different datasets. The experiments show successful results when the size of the given dataset is large.

**Key words:** Nearest neighbor rule, artificial endocrine system, condensation, consistent subset

---

### INTRODUCTION

A classification procedure is a decision-making process in which a pattern  $x_i$  of unknown class  $c$  is assigned to a possible class. To achieve optimal classifier, the procedure usually tries to minimize the classification error (Jiang *et al.*, 2009). Parametric methods assume that the conditional probability density belongs to a certain family of functions and try to get values for the parameters charactering an individual distribution. Nonparametric methods make no assumption on probability distribution and avoid the difficulties involved with the determination of multidimensional conditional probability densities. So a lot of studies are attracted within this area.

Nearest Neighbor (NN) classification is such a nonparametric method to pattern classification. The class of an unknown pattern  $x_i$  is decided by the similarities between  $x_i$  and a number of neighbors of it which lie in the neighbor area (Yang *et al.*, 2010). However, for large data sets, the cost to determine the nearest neighbors of a sample point is too expensive. It requires the storage of the complete samples and the computation of  $N$  distances if the sample size is  $N$ . So, several methods were developed to reduce the size of the sample set while maintaining the advantages of it or to reduce the amount of computation when searching for the nearest neighbor.

The original condensation method, CNN method, is proposed by Hart (1968). The objective of it is to find a consistent reference set. But the output of Hart's method depends on the order of presentation of the elements in training set  $X$ . Dasarathy (1994) represented a Minimal Consistent Set algorithm (MCS) for condensation which does not depend on the order in which the elements of  $X$  are processed. Kuncheva and Bezdek (1998) provide a counterexample to Dasarathy's conjecture that MCS method is truly minimal. Devi and Murty (2002) described a condensation algorithm, called MCNN, in which the prototypes were obtained in an incremental manner. The MCNN algorithm is order independent and many iterative calculations are needed in it to obtain a reduced set. Based on the principle of structural risk minimization, Karacali and Krim (2003) presented a fast reference set thinning method, NNSRM algorithm which was guaranteed to attain zero training error. But the cost is very high and the complexity degree of it is  $O(|T|^3)$ . Mitra *et al.* (2002) suggested a nonparametric rule reduction algorithm in which multi-scale representative fashion was used to select a subset of a large data set. Chun-Ru *et al.* (2010) proposed a two-stage condensing algorithm for KNN classifier, in which the noise data were removed firstly and the redundant samples were further eliminated by the definition of Nearest Unlike Neighbor (NUN). Other intelligent algorithms are

also used to condensate rules in training set but most of them are suitable for small or medium dataset (Cerveron and Ferri, 2001).

However, to obtain a condensed subset, iterative calculations are always needed for most of traditional algorithms. When the size of the training set is very large, most algorithms are not suitable to condense for the requirement of the storage of the whole training set and the large computation time.

In present study, we present Artificial Endocrine System based nearest neighbor rule condensation method (AES-CNN) to reduce useless points in a given training set. The method remains only the points on the boundary between different classes and the amount of reduced points of the reference set can be revised by the granularity of the lattice. The main advantages of the proposed method are, on the one hand that it is able to condense a given rule set within less time comparing with other traditional algorithms. On the other hand, with the proposed algorithm we can get a consistent subset of the given set in the divide-reduce-coalesce manner.

### ARTIFICIAL ENDOCRINE SYSTEM

Based on hormone inspired methodology, Ihara and Mori (1984) and Miyamoto *et al.* (1984) proposed an Autonomous Decentralized System (ADS), in which a content code communication protocol was used to communicate by a semantics based system. Then, Shen *et al.* (2002) described a Digital Hormone Model (DHM) in which the advantages of stochastic reasoning, distributed control, self-organization and Turing's reaction-diffusion model were integrated. The hormone model was also used by Mendao (2007) to coordinate the completion from different tasks by an autonomous robot. In his presentation, tasks were defined as glands and hormones can be released by them at a fixed speed. Once the hormones are greater than a given threshold, they are released and disseminated over the network environment. Walker and Wilson (2008) expanded the model for task assignment to a multiple-autonomous-robots environment. Each task was simply assigned to a robot and each robot can choose a suitable response to the external environment. Now-a-days, artificial endocrine systems are widely used in multiprocessor system control, real time task allocation, human-machine communication and management systems in autonomic networks.

In previous work, Xu and Lei (2011) have proposed a Lattice based Artificial Endocrine System (LAES), in

which the system was abstracted as a quintuple model,  $LAES = (L_d, EC, TC, H, A)$ :

- The environment space  $L_d$  denotes a bounded area in which all the LAES elements live, die and communicate within it.  $d$  is the number of dimensions of the space
- The endocrine cell EC is the fundamental component of the proposed model. Each endocrine cell has a perceiving sensor (AE) and a releaser (BE). The sensor AE can perceive the hormone concentrations of the lattice it lies in. The releaser BE can release a certain quantity of hormone in a discrete manner. In addition, endocrine cells were divided into different kinds. Each kind of endocrine cell can release a special kind of hormone. And every endocrine cell has several different statuses
- The target cell TC is an organ or cell that can accept stimulation from endocrine cells. The receptor AE of TC can perceive the hormone concentrations of the lattice where TC lies. Every target cell has several different statuses too
- Hormones are bio-active substances which can be used to adjust the activities of TC and EC. The cumulative hormone concentration at lattice  $L_{xy}$  is equal to the sum of the remaining concentration at time  $t-1$  and newly coming concentration at time  $t$
- Algorithm A is an iterative procedure with which all elements of LAES carry out their assigned task

### NEAREST NEIGHBOR RULE CONDENSATION

The nearest neighbor rule was originally proposed by Cover and Hart (1967). In his subsequent studies Hart (1968) represented a means for decreasing memory and computation requirements.

Assume there are  $M$  pattern classes, numbered  $1, 2, \dots, M$ . Let each pattern be defined in a  $N$ -dimensional feature space. Given  $K$  training patterns, each of them is a pair  $(x_i, c_i)$ ,  $1 \leq i \leq K$  and  $c_i \in \{1, 2, \dots, M\}$  denotes the correct pattern class and  $x_i = (x_{i1}, x_{i2}, \dots, x_{iN})$  is the set of feature values of the pattern. Let  $T_{NN} = \{(x_1, c_1), (x_2, c_2), \dots, (x_K, c_K)\}$  be the nearest neighbor training set. Given an unknown pattern  $x$ , the decision rule is to decide  $x$  is in class  $c_j$  if:

$$d(x, x_j) \leq d(x, x_i), (1 \leq i \leq K, 1 \leq j \leq K) \quad (1)$$

where,  $d(-, -)$  is some  $N$ -dimensional distance metric.

Except the excellent works done by Hart, several other promising methods were proposed. Angiulli (2007) presented a Fast Condensed Nearest Neighbor (FCNN)

rule for calculating the consistent subset of a large training set. It is order independent and requires few iteration times to converge. With this algorithm, the selected rules are very close to the decision boundary. Then, Angiuli proposed a distributed version of the FCNN algorithm to reduce the memory consumption and calculation time (Angiulli and Folino, 2007). Based on the definition of chain, Fayed and Atiya (2009) proposed a condensing method TRKNN. The basic idea of the algorithm is to separate the patterns into two different sets, condensed set and removed set, by setting a cutoff value for the chain.

**ARTIFICIAL ENDOCRINE SYSTEM FOR NEAREST NEIGHBOR RULE CONDENSATION**

The proposed algorithm remains only the points on the boundary between different classes and the amount of reduced rules of the reference set can be revised by the granularity of the lattice. The interior points of a class were seen as unimportant points which should be discarded.

The feature space of a pattern is defined as a discrete lattice environment  $L_d$ . Every rule in it is seen as an endocrine cell which has a perceiving sensor and a releaser. The two statuses of an endocrine cell are E0 and E1. In the lattice environment  $L_d$ , there is a target cell in every lattice. The three statuses of a target cell are S0, S1 and S2. Each kind of endocrine cell can secrete a special kind of hormone  $H_x$ . The only kind of hormone secreted by target cells is  $H^*$ . The hormone release procedure is a staged process. The target cells were initialized in a sleeping status S0. As long as one kind of hormone was perceived by the target cell, it will get into status S1. If two or more different kinds of hormones were perceived in the same lattice, the target cell will get into status S2. There are two steps in the algorithm:

**Step 1:** All of the rules in training set are seen as endocrine cells and were initialized in status E0. Then, the endocrine cells begin to secrete a fixed amount of hormone to the environment. The hormone will diffuse from high concentration lattices to low ones nearby

The hormone concentration  $H_x(I, j, \dots, n)_{t+1}$  of a hormone x at the lattice L  $(I, j, \dots, n)_{t+1}$  are composed of the remained concentration  $H_x(I, j, \dots, n)_t$  at time t and the diffused concentration  $\Delta H_x(i, j, \dots, n)_{t+1}$  at time t+1:

$$H_x(i, j, \dots, n)_{t+1} = H_x(i, j, \dots, n)_t + \Delta H_x(i, j, \dots, n)_{t+1}$$

$$\Delta H_x(i, j, \dots, n)_{t+1} = 1, \text{ where } H_x(i-1, j, \dots, n) >$$

$$H_x(i, j, \dots, n)_t \text{ or}$$

$$H_x(i+1, j, \dots, n)_t > H_x(i, j, \dots, n)_t \text{ or}$$

$$H_x(i, j-1, \dots, n)_t > H_x(i, j, \dots, n)_t \text{ or}$$

$$H_x(i, j+1, \dots, n)_t > H_x(i, j, \dots, n)_t \text{ or}$$

$$\dots \dots$$

$$H_x(i, j, \dots, n-1)_t > H_x(i, j, \dots, n)_t \text{ or}$$

$$H_x(i, j, \dots, n+1)_t > H_x(i, j, \dots, n)_t.$$

$$\Delta H_x(i, j, \dots, n)_{t+1} = 0, \text{ else.}$$

i, j, ..., n are coordinates of the lattice.

Having been stimulated by the hormone, the target cells changed their status according to the kind and concentration of the perceived hormones in a special lattice. When no target cell is still in status S0, the evolutionary process gets into step 2:

**Step 2:** All of the target cells in status S2 secrete hormone  $H^*$  into the lattice nearby. The concentration of it can be calculated like this:

$$C_{H^*} = T - C_{H_i} - 1$$

where,  $C_{H_i}$  is the concentration of the hormone  $H_i$  and T is the secrete time. After this, the target cells get into status S0.

The hormone  $H^*$  will diffuse from a lattice  $L^*$  to a lattice  $L^{**}$  nearby in which the target cell was still in status S1. After that, the target cell in lattice  $L^*$  gets in status S0 and the concentration of  $H^*$  in lattice  $L^{**}$  will reduce 1 unit. The diffusion process keeps on until the concentration of  $H^*$  is zero. As long as the endocrine cell perceives the hormone  $H^*$  in its lattice, the status of it will get into E1. At the end of the diffusion process, all the endocrine cells in status E1 are cells on the boundary of different classes.

**Hormone diffusion (single endocrine cell):** We can see the diffusion process from Fig. 1 where there is only one endocrine cell in the environment. At time  $t_0$ , an endocrine cell (red circle) began to secrete hormone  $H_i$ . At time  $t_1$  ( $t_1 = t_0 + \Delta t_{step}$ ), the concentration of  $H_i$  in the lattice environment was added 1 which was shown with red

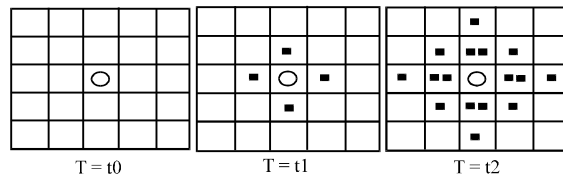


Fig. 1: Expansion from a single cell

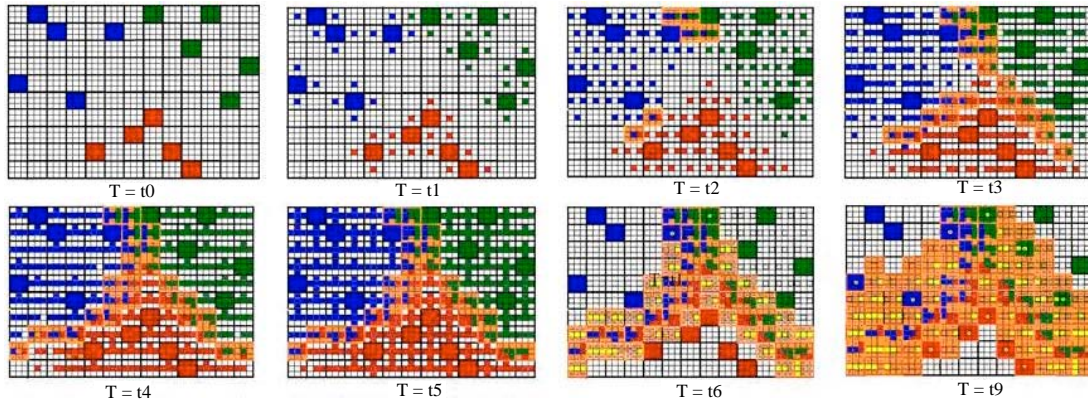


Fig. 2: Expansion from multiple endocrine cells

point.  $\Delta St_{sep}$  is the interval of time. At time  $t_k$ , the diffusion boundary  $(y_i, y_j)$  of hormone  $H_i$  secreted by endocrine cell  $(x_i, x_j)$  can be expressed by:

$$d(X, Y) = t = |x_i - y_i| + |x_j - y_j| \quad (2)$$

The area of the target cells stimulated by the hormone was expressed by:

$$\rho_p(X, Y) = \left[ \sum_{i=1}^d |x_i - y_i|^p \right]^{1/p} \quad (3)$$

When  $d = 2$  ( $d$  is the dimension number of the space) and  $p = 1$ , the red points shown the area of the target cells in status S1. So, at time  $t$ , the area of target cells controlled by an endocrine cell can be expressed by:

$$V(X) = \{Y_k | \rho_1(X, Y_k) \leq t, k \in [1-n]\} \quad (4)$$

**Hormone diffusion (multiple endocrine cells):** When the number of endocrine cells in the lattice environment is greater than 1, the evolutionary process is shown in Fig. 2, in which 15 endocrine cells (rules),  $EC_1^{(1)}$ ,  $EC_1^{(2)}$ ,  $EC_1^{(3)}$ ,  $EC_1^{(4)}$ ,  $EC_1^{(5)}$ ,  $EC_2^{(1)}$ ,  $EC_2^{(2)}$ ,  $EC_2^{(3)}$ ,  $EC_2^{(4)}$ ,  $EC_2^{(5)}$ ,  $EC_3^{(1)}$ ,  $EC_3^{(2)}$ ,  $EC_3^{(3)}$ ,  $EC_3^{(4)}$  and  $EC_3^{(5)}$  were set in the lattice. They belong to three different classes: 1, 2 and 3 (indicated by blue, red and green squares). In each lattice, there lives a target cell. At time  $t_0$ , all of the target cells (expressed by empty squares) are in status S0 and all of the endocrine cells (expressed by color square) are in status E0. At time  $t_1$ , endocrine cells secreted their hormones. The concentration of hormone in a lattice was expressed by a color point. When the target cell was stimulated by a hormone, they turned into status S1. At time  $t_2$ , 6 target cells perceived two or more kinds of hormones in their lattices and turned into status S2 (expressed by orange squares). As the time goes by, more

and more hormones were spread out. At time  $t_5$ , no target cell was in status S0. The first evolutionary stage ended.

In the second stage, all the target cells in status S2 secreted the hormone  $H^*$  (yellow points) and then turn into status S0. The hormone  $H^*$  spread out in the environment from high concentration lattices to the low one. Each time it takes place, the value of the concentration was subtracted 1. When the concentration of  $H^*$  was reduced to zero, at time  $t_9$ , the evolutionary process stopped. The endocrine cells who perceives hormone  $H^*$  in their lattices turned into status E1. And the endocrine cells in status E1 are cells we searched.

### ALGORITHM ANALYSIS

**Consistency analysis:** Let  $T_{NN} = \{(x^1, c_1), (x^2, c_2), \dots, (x^k, c_k)\}$  be the set of training samples,  $c_i (c_i \in \{1, 2, \dots, M\})$  be the class label of a sample  $(x^i, c_i)$ ,  $x^i = (x_1^i, x_2^i, \dots, x_n^i)$  be the feature set of the sample.

**Definition 1 (Sample distance):** The sample distance  $d((x^i, c_i), (x^j, c_j))$  between two samples  $(x^i, c_i)$  and  $(x^j, c_j)$  in  $N$ -dimension space is:

$$d((x^i, c_i), (x^j, c_j)) = \sum_{n=1}^N |x_n^i - x_n^j| \quad (5)$$

**Definition 2 (Consistent subset):** A consistent subset  $T_{CNN}$  of training sample set  $T_{NN}$  is a subset of  $T_{NN}$  that could classify all the patterns in  $T_{NN}$  correctly.

**Definition 3 (Minimal consistent subset):** The minimal consistent subset  $T_{MCNN}$  is the smallest and most efficient subset of  $T_{NN}$  that could properly classify every pattern in  $T_{NN}$ .

**Theorem 1:** The subset  $T_{AES-CNN}$  of a given sample set  $T_{NN}$ , obtained by algorithm AES-CNN, is a consistent subset of  $T_{NN}$ .

**Proof:** Let  $T_{NN1}$  be the set of endocrine cells which are in E1 status at the end of the second stage. Let  $T_{NN2}$  be the set of endocrine cells which are in E0 status at the end of the second stage.

According to algorithm AES-CNN, the training sample set  $T_{NN}$  is composed of  $T_{NN1}$  and  $T_{NN2}$ :

$$\begin{aligned} T_{NN2} &= \{(x^{j1}, c_{j1}), (x^{j2}, c_{j2}), \dots, (x^{jn}, c_{jn})\} \\ T_{NN1} &= \{(x^{i1}, c_{i1}), (x^{i2}, c_{i2}), \dots, (x^{im}, c_{im})\} \\ T_{NN} &= T_{NN1} \cup T_{NN2} \end{aligned}$$

Let  $(x^j, c_j)$  be a cell in  $T_{NN2}$ . Let  $(x^*, c^*)$  be the nearest cell of cell  $(x^j, c_j)$  and the two cells have the same class label. Let  $(x\&, c\&)$  be the nearest cell of cell  $(x^j, c_j)$  and the two cells have the different class label. Thus:

$$\begin{aligned} d((x^*, c^*), (x^j, c_j)) &\leq d((x^i, c_i), (x^j, c_j)) \quad (c^* = c_j) \text{ and } (c_i = c_j) \\ d((x\&, c\&), (x^j, c_j)) &\leq d((x^i, c_i), (x^j, c_j)) \quad (c\& \neq c_j \text{ and } c_i \text{ and } c_j) \end{aligned}$$

If we want to show that  $T_{NN1}$  is a consistent subset of  $T_{NN}$ , the only task for us is to show that for every cell  $(x^j, c_j)$  in  $T_{NN2}$ , the assertion below is true:

$$d((x^*, c^*), (x^j, c_j)) \leq d((x\&, c\&), (x^j, c_j))$$

**Assume:**

$$d((x^*, c^*), (x^j, c_j)) > d((x\&, c\&), (x^j, c_j))$$

Then, according to rules in step 1, there will be a target cell  $TC^k$  which was stimulated by the two kind of hormones released by cell  $(x\&, c\&)$  and cell  $(x^j, c_j)$ .

Thus, in the second stage, the  $TC^k$  will release hormone  $H^*$ . And then, having been stimulated by hormone  $H^*$ , the cell  $(x^j, c_j)$  will come into state S0. So, the cell  $(x^j, c_j)$  should belong to  $T_{NN1}$ .

However, it is a contradictory assertion against the assumption  $(x^j, c_j) \in T_{NN2}$ .

So, we can get  $d((x^*, c^*), (x^j, c_j)) \leq d((x\&, c\&), (x^j, c_j))$ .

**Complexity analysis:** Complexity of an algorithm refers to the amount of time and space required to execute the algorithm in the worst case. AES-CNN algorithm is a parallel algorithm. If we implement it in a series manner, the time needed to execute it comprises with two main parts:

- In the first stage: the N coordinates of the N-dimension space are divided into  $n_1, n_2, \dots, n_N$  uniform lattices. The endocrine cells (sample rules) were

placed in the lattice environment and released a certain amount of hormones at every discrete time. In the worst cases, these can be performed in  $O(N \times n_1 \times n_2 \times \dots \times n_N \times n^*)$  time.  $n^*$  is the maximal elements in the set  $\{n_1, n_2, \dots, n_N\}$

- In the second stage: the target cells were stimulated and released hormone  $H^*$ . The concentration of  $H^*$  in a lattice was subtracted 1 every discrete time. These can be performed in  $O(N \times n_1 \times n_2 \times \dots \times n_N \times n^*)$  time

By summing up the computational time required for each of the two stages, it is possible to determine the total computational time of the algorithm. The computational time of the whole process is given by:

$$\begin{aligned} O(n1 \times n2 \times \dots \times nN \times n^*) + O(n1 \times n2 \times \dots \times nN \times n^*) \\ = O(2 \times n1 \times n2 \times \dots \times nN \times n^*) \end{aligned} \quad (6)$$

It is clear that the computational time of the algorithm is determined mainly by the number of the dimension, the numbers of the discrete lattices in each coordinates axis.

The memory required to run the algorithm is proportional to the number of samples K, dimensions N and the number of discrete lattices.

## EXPERIMENTAL RESULTS

The experiments were conducted on a 3.0 GHz Pentium4 with 1.0 GB of memory running Microsoft Windows XP. All code was compiled using Microsoft Visual C++6.0. The datasets used in the experiment are obtained from the UCI Machine Learning Repository (Newman *et al.*, 1998).

**Data preprocessing:** In AES-CNN algorithm, the lattice granularity is an important parameter. Too small size of it will lead to a slow implementation. So, in order to speed up the implementation process, we preprocess the training set like this:

Let  $Z(I, j, \dots, n)$  be the location of an endocrine cell and  $i, j, \dots, n$  are coordinates. In general, the values of them are always real number. In order to assign a suitable value for each of them, we give several preprocessing operations:

- Rank the endocrine cells according to their coordinate  $i$ , the former is the smaller. And then, assign new value for each of this coordinate  $i$  like this: 1, 2, ..., M
- Rank the endocrine cells according to their coordinate  $j, \dots, n$  and assign new value for each of the coordinate just like it in (step 1)

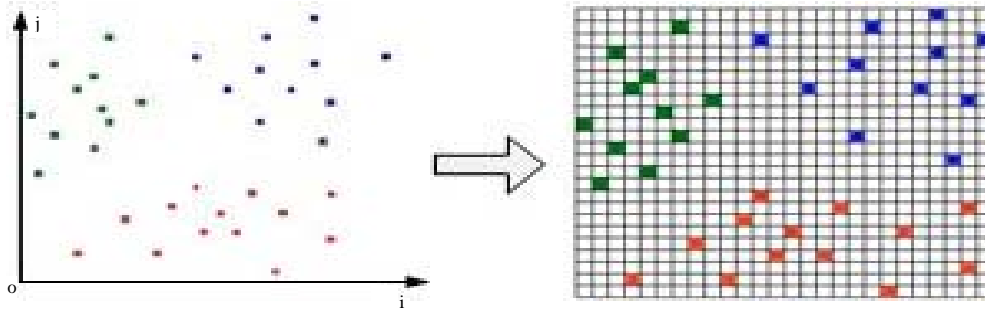


Fig. 3: Data preprocessing

Table 1: Different granularities (Iris)

Grain size	Class	Training set	Test set	Deleted rules				Condensed rules	Accuracy	Time
				A	B	C	D			
37*25*60*25	3	120	30	1	0	0	58	61	29/30	986s
19*13*30*13	3	120	30	1	3	0	39	77	29/30	35s
10*7*15*7	3	120	30	1	36	0	14	69	29/30	1s
5*4*8*4	3	120	30	1	72	7	4	36	28/30	<1s
3*2*4*2	3	120	30	1	63	38	1	17	26/30	<1s
Preprocessing	3	120	30	1	0	0	50	69	29/30	1s

After doing so, we can see the endocrine cells in the lattice environment from Fig. 3.

**Lattice granularity:** Here, the data set used is Iris. There are 150 instances, 4 condition attributes and 1 class attribute in it. We divided the data set into two parts: training set (120 instances) and test set (30 instances). With the proposed AES-CNN algorithm, we delete some rules in training set and obtained the consistent subset of it. From Table 1, we can see that the numbers of the deleted rules and remained rules, the accuracy of the classifier built by the remained rules and the time needed are listed in column 5, 6, 7 and 8. We divided the 4 condition attributes with different grain sizes (we can see it from column 1). That leads to different number of lattices in the artificial endocrine environment.

From Table 1, we can see that as the increasing of the grain size of the lattice, the number of the lattices becomes smaller and the running time of the algorithm gets faster. However, the accuracy of the algorithm becomes more less.

When the value of column 5-C is 0, we can see that the accuracy of the record 1-3 keep unchanged, although the grain size was enlarged. But for record 4-5, the accuracy of it declined. The reason is that the decline of the accuracy of the algorithm mainly decided by the grain size of the lattice. When different classes of cells were swallowed in the initial step, the algorithm can not find the right boundary in the evolution stage. For example, in record 4 and 5, the numbers of the swallowed rules of different class labels were 7 and 38. So, the unsuitable

grain size of the lattice will lead to incorporation of the rules in different class. As a result, the classification accuracy will decline. However, after preprocessing, the algorithm will find a classifier faster on the case of similar accuracy.

**Different instance size:** Here, the data sets used are Car and Transfusion. There are 1728 records and 4 class labels in dataset Car and 748 records and 2 class labels in dataset Transfusion. We randomly divided the two data sets into two parts: one for training (1530, 592 instances), one for test (198, 156 instances). Then, we choose rules from training set to build 9 different sub-training sets. The instance numbers of the sub-training sets in Car are 170, 340, ..., 1530, in Transfusion are 74, 148, ..., 592.

From Fig. 4 we can see that with the increasing of the number of the training instances, the accuracies of the two algorithms AES-CNN and algorithm CNN [3] improve simultaneously. However, when the number of the training instances is small, the accuracy of algorithm CNN is greater than AES-CNN. When the number of the training instances gets large, the accuracy of the two algorithms are almost the same. It is because that the remained rules obtained by AES-CNN are boundary cells but by CNN, they are cells selected randomly. When the boundaries between different classes are complex, more boundary cells were remained and the accuracy of the algorithm AES-CNN is low and unstable. We can see it from Fig. 4. For dataset Car, the accuracy of AES-CNN is lower than CNN and the number of the rules remained in AES-CNN is larger than it in CNN.

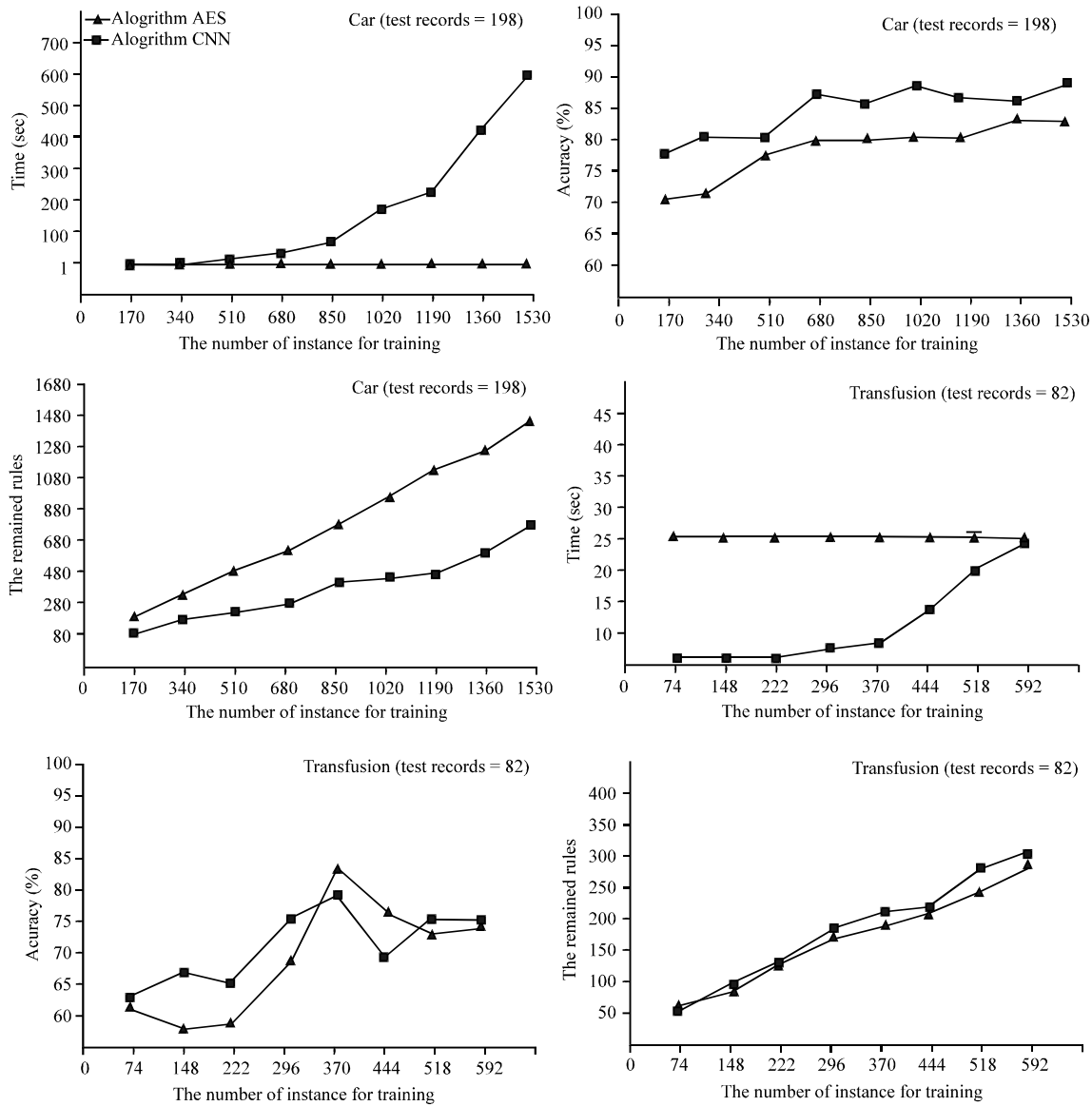


Fig. 4: Different sizes of training set

However, for dataset Transfusion, the accuracies of the two algorithms and the remained rules of them are very similar. Why? The reason is that for dataset Transfusion, the numbers of the feature attributes and classes are small. The boundary of the dataset is very simple. So, the boundaries between different class cells can be recorded by fewer boundary cells.

Comparing with the accuracy and the number of remained rules in the two algorithms, the runtime of the two algorithms are so different. For both of them, with the increasing of the training instances, the time cost by CNN increased quickly. But the time cost by AES-CNN is fixed.

The difference between the two datasets is that, for Car, the runtime of AES-CNN is less than CNN. For Transfusion, the runtime of AES-CNN is more than CNN.

We can find the reason from formula (6) which shows that the computational time of CNN is  $O(K!)$  while for AES-CNN, the time is  $O(2 \times N \times n_1 \times n_2 \times \dots \times n_N \times n^*)$ . For a given dataset, when the dimensions and grain size of lattice are fixed, the running time of algorithm AES-CNN is fixed. When the number of the records is small and the attributes of it is large, the efficiency of algorithm CNN will greater than AES-CNN. Otherwise, the efficiency of algorithm AES-CNN will greater than CNN.



Table 2: Performance results on different datasets

Dataset	AES-CNN				CNN			TRKNN			FCNN		
	TRAI No.	ACCU# (%)	STOR# (%)	Time# (sec)	ACCU# (%)	STOR# (%)	Time# (sec)	ACCU# (%)	STOR# (%)	Time# (sec)	ACCU# (%)	STOR# (%)	Time# (sec)
Diagnosis	96	100	85.4	<1	100	11.5	<1	100	10.1	<1	100	10.4	<1
Iris	120	96.7	57.5	<1	95.6	13.3	<1	94.0	11.7	<1s	93.7	12.5	<1
Tae	121	66.7	76.0	42	66.7	71.9	1	66.9	68.8	<1	66.7	70.2	<1
Monks_1	122	89.2	73.0	<1	86.6	36.9	<1	86.1	35.1	<1	87.5	36.1	<1
Monks_2	124	86.7	74.2	<1	85.6	46.8	<1	85.2	43.4	<1	86.1	45.9	<1
Monks_3	169	78.2	81.1	<1	77.1	68.0	<1	76.3	63.7	<1	77.8	65.7	<1
Haberman	245	65.5	81.2	6	60.7	46.1	1	60.1	43.2	<1	63.9	44.1	<1
Ecoli	269	83.4	64.3	14	79.1	36.1	4	78.6	34.5	2	82.1	35.3	1
Balance-scale	500	86.2	43.2	<1	84.0	40.6	15	84.1	38.5	8	85.6	39.8	3
Transfusion	599	79.5	28.4	18	77.2	49.4	29	77.1	47.2	12	78.5	48.6	5
Pima-indians	615	67.6	59.6	17	64.7	50.9	55	64.3	47.7	31	66.7	49.8	8
Cmc	1179	49.1	63.5	78	45.6	73.4	507	45.9	71.6	268	47.0	72.3	66
Car	1383	91.2	82.4	21	88.4	41.6	430	88.0	40.0	219	89.6	40.3	58

Table 3: Consistency comparison-3

Dataset	AES-CNN			CNN			FCNN			TRKNN		
	CONS# (%)	RULE# (%)	Time# (sec)	CONS# (%)	RULE# (%)	Time# (sec)	CONS# (%)	RULE# (%)	Time# (sec)	CONS# (%)	RULE# (%)	Time# (sec)
Diagnosis	88.5	86.2	<1	74.2	15.5	<1	80.2	14.3	<1	73.6	16.7	<1
Iris	86.7	61.2	<1	66.7	21.0	<1	71.0	18.6	<1	67.1	22.5	<1
Tae	76.7	75.5	20	52.7	57.9	<1	56.7	51.9	<1	51.9	58.6	<1
Monks_1	84.8	74.2	<1	66.6	42.9	<1	67.5	38.2	<1	64.5	44.8	<1
Monks_2	85.1	75.5	<1	75.6	59.8	<1	76.1	57.7	<1	74.2	55.4	<1
Monks_3	86.2	82.0	<1	62.1	68.0	<1	66.8	66.2	<1	60.5	66.7	<1
Haberman	88.7	81.2	3	60.7	49.1	<1	63.9	45.5	<1	60.1	50.2	<1
Ecoli	86.8	74.3	6	77.1	46.1	2	79.0	41.1	<1	74.6	47.8	1
Balance-scale	93.0	83.2	<1	73.0	46.6	5	77.6	40.2	2	71.2	46.5	3
Transfusion	90.2	79.6	8	71.2	52.8	4	72.5	49.1	2	70.8	53.2	2
Pima-indians	86.0	81.6	8	62.7	51.9	20	66.7	50.2	3	63.0	52.8	11
Cmc	82.8	83.5	37	52.6	76.4	337	55.0	63.4	36	51.5	77.7	187
Car	84.8	82.4	8	69.6	47.1	76	72.2	42.5	9	67.8	47.8	39

**Comparisons with different datasets:** In Table 2, we compared the proposed AES-CNN algorithm with CNN, TRKNN and FCNN algorithms on 13 datasets coming from UCI Machine Learning Repository. Each dataset was divided into two parts: training set and test set. The size of the train set, the percent of the accuracy and storage of the classifier (comparing with no condensate one) and the runtime of the algorithm are shown in Table 2.

As can be seen, as the size of the training set getting larger, AES-CNN showed more feasible on runtime. For example, on dataset Car, the runtime of AES-CNN is less than 1 sec. However, for CNN, it is 430 sec. But the efficiency of AES-CNN is constrained by the grain size of the lattice. When there are too many lattices in the AES circumstance, AES-CNN will cost more runtime. For example, on dataset Tae, the runtime of AES-CNN is 42 seconds which is much longer than that of algorithm CNN.

In general, the accuracy of the classifier obtained by algorithm AES-CNN is higher than others. But the difference is not very obvious. The numbers of the rules used to build the classifier are very different. For AES-CNN, more remained rules were left. The reason of that is the nature of the algorithm AES-CNN is to find the

boundary cells, the number of which is generally larger than the number of the remained rules gotten by CNN or other algorithms when the distributions of different classes in a given dataset are very complex.

With the comparison above we can find that the algorithm AES-CNN is more suitable for the case where the size of the dataset is very large. When the boundary between different classes is very complex, the size of the condensate subset gotten by AES-CNN is larger than that gotten by other algorithms. On the contrary, the size is less than others.

**Consistency comparison (divide-reduce-coalesce manner):**

Here, we check the consistency of the proposed algorithm with divide-reduce-coalesce manner. The two datasets used in this section are Car and Transfusion. In step 1, we randomly split an original set A0 into four subsets, A1, A2, A3 and A4. Then, we condensed the four subsets, respectively with different algorithms and obtained four sets, A1', A2', A3' and A4'. In step2, we obtained set B1 by integrating the two sets A1' and A2' and obtained set B2 by integrating A3' and A4'. In step3, we condensed the two sets B1, B2 and obtained two subsets B1' and B2'. Finally, we integrate the sets B1', B2'

and condensed them to get the terminal set C. The original set A0 was seen as the test set.

From Table 3, we can see that after the divide-reduce-coalesce condensation, all of the final four subsets gotten by four different algorithms are not consistent sets of the original given set A0. However, the accuracy of algorithm AES-CNN is far higher than algorithms CNN, TRKNN and FCNN. And the runtime of algorithm AES-CNN in such a manner is also superior to other algorithms.

The reason why the subset C obtained by algorithm AES-CNN is not a consistent subset of A0 lies in the manner of data preprocessing and the manner of distance calculation. The distances between different points, calculated with discrete coordinates, are not accurate ones. Even so, from the experiments we can see that the AES-CNN algorithm is more suitable for divide-reduce-coalesce learning than other algorithms.

## DISCUSSION AND CONCLUSION

The algorithm presented in this study is an artificial endocrine system based nearest neighbor rules condensation approach. The main advantages of this method are, on the one hand that it is able to condense a given rule set within less time when the number of the rules in a given rule set is very large. On the other hand, with the algorithm, we can get a consistent reference set of the given set in an iterative or accumulated manner.

To achieve this, an artificial endocrine model was developed so as to a parallel complementation of the condensation process. Previous works pay more attentions to the accuracy of the classifier built by the remained rules. But the implement time increases very quickly as the rule number becomes more and more large. However, in this study we pay more attentions to the reduction of the runtime and the capability of the accumulated running. The method remains only the points on the boundary between different classes. The algorithm can guarantee little resubstitution error and at the same time, the generalization capability of the nearest prototype classifier is considered.

## ACKNOWLEDGMENT

We are grateful to the anonymous referees for their invaluable suggestions to improve the study. This work was supported by the National Natural Science Foundation of China (60873035).

## REFERENCES

Angiulli, F. and G. Folino, 2007. Distributed nearest neighbor-based condensation of very large data sets. *IEEE Trans. Knowledge Data Eng.*, 19: 1593-1606.

- Angiulli, F., 2007. Condensed nearest neighbor data domain description. *IEEE Trans. Pattern Anal. Machine Intell.*, 29: 1746-1758.
- Cerveron, V. and F.J. Ferri, 2001. Another move toward the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule. *IEEE Trans. Syst. Man Cybernetics Part B*, 31: 408-413.
- Chun-Ru, D., P.P.K. Chan, W.W.Y. Ng and D.S. Yeung, 2010. 2-Stage instance selection algorithm for KNN based on Nearest Unlike Neighbors. *Proceedings of the International Conference on Machine Learning and Cybernetics*, July 11-14, Qingdao, pp: 134-140.
- Cover, T. and P. Hart, 1967. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theor.*, 13: 21-27.
- Dasarathy, B.V., 1994. Minimal Consistent Set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Trans. Syst. Man Cybernet*, 24: 511-517.
- Devi, V.S. and M.N. Murty, 2002. An incremental prototype set building technique. *Pattern Recognition*, 35: 505-513.
- Fayed, H.A. and A.F. Atiya, 2009. A novel template reduction approach for the k-nearest neighbor method. *IEEE Trans. Neural Networks*, 20: 890-896.
- Hart, P.E., 1968. The condensed nearest neighbor rule (Corresp.). *IEEE Trans. Inform. Theory*, 14: 515-516.
- Ihara, H. and K. Mori, 1984. Autonomous decentralized computer control systems. *IEEE Comput.*, 17: 57-66.
- Jiang, L., H. Zhang and Z. Cai, 2009. A novel bayes model: Hidden naive bayes. *IEEE Trans. Knowledge Data Eng.*, 21: 1361-1371.
- Karacali, B. and H. Krim, 2003. Fast minimization of structural risk by nearest neighbor rule. *IEEE Trans. Neural Networks*, 14: 127-134.
- Kuncheva, L.I. and J.C. Bezdek, 1998. On prototype selection: Genetic algorithms or random search. *IEEE Trans. Syst. Man Cybernet*, C28: 160-164.
- Mendao, M., 2007. A neuro-endocrine control architecture applied to mobile robotics. Ph.D. Thesis, University of Kent, Canterbury.
- Mitra, P., C.A. Murthy and S.K. Pal, 2002. Density-based multiscale data condensation. *IEEE Trans. Pattern Anal. Machine Intell.*, 24: 734-747.
- Miyamoto, S., K. Mori, H. Ihara, H. Matsumaru and H. Ohshima, 1984. Autonomous decentralized control and its application to the rapid transit system. *Int. J. Comput. Ind.*, 5: 115-124.
- Newman, D.J., S. Hettich, C. Blake and C. Merz, 1998. UCI Repository of Machine Learning Databases. Dept. Information Computer Science, University of California, Berkeley, CA.

- Shen, W.M., B. Salemi and P. Will, 2002. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Trans. Robotics Automation*, 18: 700-712.
- Walker, J. and M. Wilson, 2008. A performance sensitive hormone-inspired system for task distribution amongst evolving robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 22-26, New York, pp: 1293-1298.
- Xu, Q.Z. and W. Lei, 2011. Lattice-based artificial endocrine system model and its application in robotic swarms. *Sci. China Inform. Sci.*, 54: 795-811.
- Yang, J.M., P.T. Yu and B.C. Kuo, 2010. A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. *IEEE Trans. Geosci. Remote Sensing*, 48: 1279-1293.