A Virus Detection System Based on Artificial Immune System

Rui Chao and Ying Tan, *Member, IEEE*

***Abstract*— A virus detection system (VDS) based on artificial immune system (AIS) is proposed in this paper. VDS at first generates the detector set from virus files in the dataset, negative selection and clonal selection are applied to the detector set to eliminate autoimmunity detectors and increase the diversity of the detector set in the non-self space respectively. Two novel hybrid distances called hamming-max and shift r bit-continuous distance are proposed to calculate the affinity vectors of each file using the detector set. The affinity vectors of the training set and the testing set are used to train and test classifiers respectively. VDS compares the detection rates using three classifiers, k-nearest neighbor (KNN), RBF networks and SVM when the length of detectors is $32-bit$ and $64-bit$. The experimental results show that the proposed VDS has a strong detection ability and good generalization performance.**

I. INTRODUCTION

The analysis of behavior and binary data of programs are the two main methods employed by an anomaly detection system to recognize malicious programs.

Behavior-based methods utilize the operating system's application programming interface (API) sequences, system calls or other kinds of behavior characteristics to identify the purpose of a program [1]. These methods at first construct profiles during the legitimate operations of the monitored programs. During the detection process, any system call sequence or argument that do not comply with the previously generated 'normal' profiles is regarded as a sign that the system is compromised. The corresponding program will be stopped and then classified as a malicious. The malicious are identified when the computer are already damaged, so many methods use a virtual environment (called sandbox) to simulate real system environment where the unclassified programs are running. The drawbacks of these methods cost much time and effort to build a less-error sandbox, but the sandbox can not simulate the same environment as real operating system. Although these approaches have produced promising results, they can produce high rates of false positive errors, an issue which has yet to be resolved [2].

Data-based methods can detect virus before they are executed, they utilize the binary data extracted from the program files. The traditional methods extract signatures from virus samples [3], scanners compare these signatures with unclassified files to determine whether they are virus. These methods are effective in the past. As novel advanced technologies are widely used in manufacturing new virus, polymorphic virus can change their signatures while spreading. So it is getting harder both for experts to extract signatures and to detect them.

This leads to some heuristic data-based methods. Among those methods, AIS as a dynamic, adaptive, distributed learning system are applied to many fields of science and engineering, such as pattern recognition, fault diagnosis, network intrusion detection and virus detection. There are three basic models based on AIS principles: negative selection algorithm, clonal selection algorithm and the immune network model. It has a great similarity to computer security system. The concepts, self, non-self and so on, can be applied to the computer security system as well. Therefore, virus detection using AIS has paved a new way for anti-virus research.

Authors are with Key Laboratory of Machine Perception, Ministry of Eduction, Peking University, and with Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, P.R. China. Prof. Y. Tan is the corresponding author. ytan@pku.edu.cn

II. RELATED WORK

Some statistical or heuristic algorithms utilize binary data to mine the characteristics of unclassified programs. Among these algorithms, the self and non-self hypothesis [4] in AIS are proposed for years. The theoretical foundations of the distribution change detection methods [5] are verified based on that hypothesis. A number of work have been done to design an effective virus detection system. The algorithm in [6] at first generates the detectors from short executable binary sequences, then uses a NN-based classifier to discriminate malicious and benign executables. This algorithm uses five detector sets with different length of detectors and BP network to train the classifier, so its large computational cost and long time training time of BP network leads to weak performance when the size of the file increases dramatically.

Aiming at building a light-weighted, limited computer resource and early virus warning system (VDS), a novel virus detection system on the basis of AIS was proposed in this paper. As binary data of files are directly utilized by VDS to generate a detector set, no sandbox or signature extraction processed are needed.

The difficult part in building VDS is that there doesn't exist standard virus database available for testing and comparing of algorithms. So many algorithms were proposed just based on a few specific virus, and some of them have a very poor generalization ability. As a result, we collect as much virus samples as possible for testing VDS using different percentages of files in a dataset to enhance the generalization ability for a high accuracy for new viruses.

TABLE I

THE NUMBER OF FILES IN EACH DATASET IN OUR EXPERIMENT

	Training Set		Testing Set	
Data sets	Benign Files	Virus Files	Benign Files	Virus Files
Dataset1	71	885	213	2662
Dataset2	142	1773	142	1774
Dataset3	213	2662	71	885

III. THE PROPOSED VDS

A. Dataset

There are 284 benign files with $78MB$ in total, and 3547 virus files with $7.8MB$ in the data set (DS). Another data set contains 208 benign files is used for negative selection, totally $189MB$. All the benign files are system files or well-known programs with extensions .exe.

The DS is randomly divided into different percentages of training set and testing set as shown in table I, there is no overlap in the two sets.

B. Sliding window

Bit slices in a fixed length $L - bit$ (binary data fragments) x_j^f are extracted from each file using a sliding window. The set of x_j^f of file l is represented by $DF_l = \{x_0^f, x_1^f, ..., x_{n-1}^f\}$, $|DF_l| = n^f$, $|x_j^f| = L$. Two neighboring fragments have an overlap of $\lfloor L/2 \rfloor$ bits. If the size of the file l is N, $|DF_l| = 2N/L$. DF_l is the all information utilized by VDS. The process of extracting data fragments from a binary file is shown in Fig. 1.

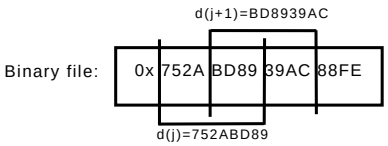

Fig. 1. The process of extracting data fragments, $|x_j^f| = L = 32$ bits.

In the hypothesis of immune system, antibody T-cells detect antigens using the information of protein portions on the surface of antigen [7] [8]. In VDS, x_j^f is corresponding to the protein portions of an antigen, an entire program file is regarded as an antigen.

C. Negative selection

In AIS, as the T-cells mature in the thymus, they undergo a censoring process called negative selection, in which those T-cells that bind self cells are destroyed [9] [10]. After censoring, T-lymphocytes that do not bind self are released to the rest of the body and provide a basis for our immune protection against foreign antigens. This mechanism in the immune system is very robust because of its distribution nature and high efficiency.

DTI is the data fragment set extracted from virus files in the training set, it contains immature detectors which will undergo negative selection and clonal selection before they are used to detect virus. Since most viruses insert or append themselves to benign files, DTI contains both benign and virus data fragments which will lead to false positive during detection. The purpose of the negative selection is to remove x_j^f appears both in DTI and benign files, only virus-special data fragments are left in DTI. The process of negative selection is shown in Fig. 2.

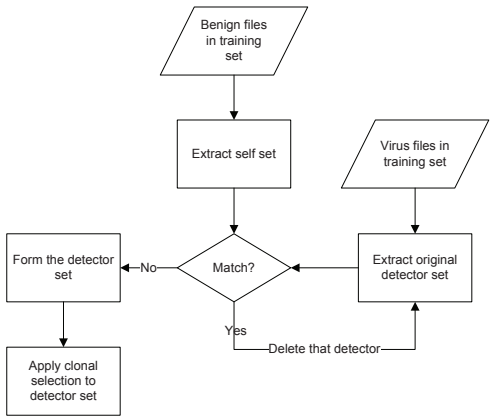

Fig. 2. Negative selection process

D. Clonal selection

The clonal selection algorithm is used by AIS to define the basic features of an immune response to an antigenic stimulus [11] [12]. It establishes the idea that only those cells that recognize the antigens are selected to proliferate. The selected cells are subject to an affinity maturation process, which improves their affinity to the selective antigens.

Clonal selection is used to increase the diversity of DTI in the non-self space, new detectors are generated from data fragments in DTI. The newly generated detectors would not only cover as much non-self space as possible but also enhance the ability of detecting unknown viruses. The number of clones generated is given by:

$$|C(x_i^t)| = \frac{\alpha}{F_{x_i^t}}, \tag{1}$$

where α is the coefficient of clone selection, usually $\alpha = 10$. x_i^t represents a single detector in DTI and $F_{x_i^t}$ is the occurrence frequency of x_i^t, $C(x_i^t)$ is number of clones generated by x_i^t.

Traditional AIS mutation methods are used in clonal selection, at most 5 bits of x_i^t are randomly changed to generate new detectors. The lower $F_{x_i^t}$ is, the higher mutation rate will be taken.

E. Distances

After negative selection and clonal selection are applied to DTI, the detectors in DTI are matured, so it can be used as the detector set DT to detect virus. In VDS, the affinity value reflects the similarity between a data fragment from an unknown file and the virus. To calculate the affinity values between x_j^f in DF_l of file l and x_i^t in DT, two novel distances and shift operator introduced to efficiently

detect short non-continuous but virus-special assemble code instructions.

1) Hamming-max distance: Hamming distance [13] plus cyclic shift operator(called hamming-max distance) is used to find the best matching position between $x_j^f \epsilon DF_l$ and $x_i^t \epsilon DTI$ during the matching process. Hamming-max distance can be evaluated by the following equation:

$$HM(x_i^t, x_j^f) = max\{HD(x_{i'}^t, x_j^f)\}, \tag{2}$$

where $HM(x_i^t, x_j^f)$ and $HD(x_i^t, x_j^f)$ are the hamming-max and hamming distance between x_i^t and x_j^f respectively. $x_{i'}^t \epsilon \{S(x_i^t, 0, left),\ S(x_i^t, 1, left),\ ...,\ S(x_i^t, L-1, left)\}$, $|x_{i'}^t| = L$, $S(x_i^t, k, left)$ means left cyclic shift x_i^t k bits.

Hamming-max distance can avoid the influence of bits mismatching to enhance the ability of matching exactly the virus-special instructions.

2) Shift r-continuous bit distance: R-continuous bit distance [14] [15] is widely used in binary string matching. If two strings contain the same substring with the length of r, they are matched. Shift r-continuous bit distance is aimed to detect shorter serial assemble instructions which appear rarely in benign programs. It is a supplement to the hamming-max distance. The shift r-continuous bit distance $SR(x_i^t, x_j^f, r)$ between x_i^t and x_j^f can be evaluated by the following equation:

$$SR(x_i^t, x_j^f, r) = max\{R(x_{i'}^t, x_j^f, r)\}, \tag{3}$$

where $R(x_{i'}^t, x_j^f, r)$ is the r-continuous bit distance with the length of matching r bits. $x_{i'}^t \epsilon \{S(x_i^t, 0, left), S(x_i^t, 8, left), ..., S(x_i^t, L-8, left)\}$. The lengths of assemble instructions vary from 8 to 64 bits and shorter instructions appears more frequently than longer ones in most cases, so the VDS select $r = 12\text{-}bit$ and $r = 24\text{-}bit$ repectively. The length of shift used by shift operator is $8\text{-}bit$ long, it is the minimum length of an assemble instruction.

F. Affinity vector

Based on the "immune ball" theory, an antibody has its limited detection space, antigens in that space has closer distance to it than to other antibodies. In VDS, detectors having the identical last $|K|$ bits (K is called the index bits) to data fragment x_j^f are assumed to be neighbors in the detection space of x_j^f. $DTS_{x_j^f}$ represents the sub detector set of DT, the danger level of x_j^f can be calculated by the following equation:

$$DL(x_j^f) = \frac{\sum_{i=0}^{n^f} \langle HM(x_i^t, x_j^f), SR(x_i^t, x_j^f, 12), SR(x_i^t, x_j^f, 24)\rangle}{n^f} \tag{4}$$

where $x_i^t \epsilon DTS_{x_j^f}$, detectors in $DTS_{x_j^f}$ have identical $|K|$ bits with x_j^f. $DL(x_j^f)$ is the danger level of x_j^f in DF_l of file l, $|DF_l| = n^f$.

The average affinity value of x_j^f in DF_l is the dangerous level of file l. It can be obtained by the following equation:

$$v_l = \frac{\sum_{j=0}^{n^f} DL(x_j^f)}{n^f}, \quad |DF_l| = n^f, \tag{5}$$

where v_l is the affinity vector of file l, it reflects the dangerous level of file l from three different distances. The higher v_l is, the more likely file l is a virus.

The calculation process of the dangerous level of $file_l$ is illustrated as follows.

1. Extract the data fragment set DF_l from file l, it is a multi-set because some data fragments may have the same value. $x_j^f \epsilon DF_l$, $|x_j^f| = L$ bits;

2. Calculate v_l using distances described in III-E. The pseudo code is shown in **Algorithm 1**.

Algorithm 1 Calculation of v_l

```
v_l = 0;
for every x_j^f in DF_l do
    v_l' = 0;
    if DTS_{x_j^f} is empty then
        Continue;
    end if
    for every x_i^t in DTS_{x_j^f} do
        v_l' += ⟨HM(x_i^t, x_j^f), SR(x_i^t, x_j^f, 12), SR(x_i^t, x_j^f, 24)⟩;
    end for
    v_l += v_l' / |DTS_{x_j^f}|;
end for
v_l \= |DF_l|;
```

If $DTS_{x_j^f}$ is empty, 0 is signed to x_j^f. The last part of **Algorithm 1** is a normalization process.

G. Training classifiers with affinity vectors

Three classifiers, RBF network, SVM [16] [17] with a radial basis kernel function(rbf-SVM) and KNN, are used to detect the viruses in the testing set for comparison. They are trained with affinity vectors of files in training set evaluated in III-F. The entire process of the VDS is shown in Fig. 3.

IV. Experiments

A. Length of data fragment

The length of data fragments L is critical to VDS. If L is too short, the d_j^f can not contain enough virus-special information for detecting, the space 2^L is too small to discriminate self and non-self; if L is too long, the space 2^L is too large that every x_j^f is rare in the space and x_j^f contains too much information that makes virus-special data hidden in DF_l. In each condition above, the generalization ability of the VDS will be limited.

As the length of a single assemble code instruction varies from 1 byte to 7 bytes, L is not necessary bigger than 64-bit to contain at least one entire instruction. So the experiments choose $L = 64$-bit and $L = 32$-bit. The overlap of x_j^f is always $\frac{L}{2}$.

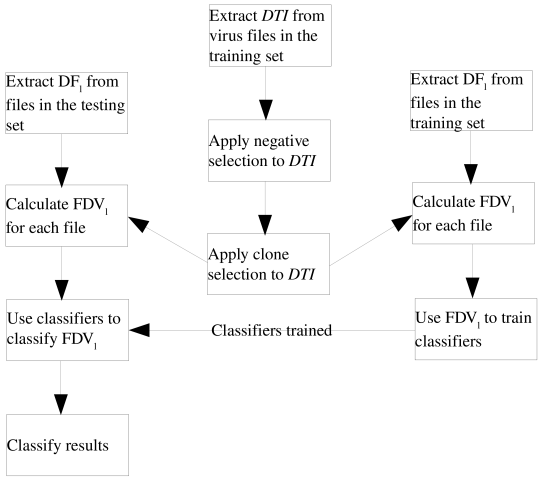

Fig. 3. The process of VDS. (a) Extract data fragments DF_l from virus files in the dataset, apply negative selection and clonal selection to DTI to form the detector set DT. (b) Use DT to calculate the dangerous level of all files v_l in the dataset. (c) Train classifiers using v_l in the training set. (d) Classify v_l in the testing set.

TABLE II

THE AVERAGE DETECTION RATE OF SVM WHEN $L = 32$ AND $L = 64$, THE FILES ARE RANDOMLY SELECTED FROM THE DATASET

Detection Rates		$L = 32$		$L = 64$	
Database		**Virus**	**Benign Files**	**Virus**	**Benign Files**
Data Set1	Training Set	99.55%	97.18%	100%	97.18%
	Testing Set	91.28%	99.06%	84.44%	99.53%
Data Set2	Training Set	99.38%	98.59%	100%	97.18%
	Testing Set	92.45%	98.59%	89.06%	97.89%
Data Set3	Training Set	99.21%	99.06%	100%	99.53%
	Testing Set	93.46%	95.77%	89.06%	97.18%

B. Results

The detection rates using SVM classifier with different length of L are shown in table II. It shows similar performance on all the data sets. 32-bit detectors have better accuracy and generalization ability in detecting virus in the testing sets. So $32 - bit$ data fragments contain enough virus characteristics information for detection, and $64 - bit$ data fragments contain too much benign codes, reducing the thickness of virus information.

The comparison experimental results are shown in Fig. 4 and Fig. 5.

Figure 4 and 5 shows that RBF network has better performance than SVM and KNN except on virus files in testing set. But it has weaker generalization ability especially when the training data is small, even though it has the highest detection rate on all other data sets. KNN and SVM have nearly the same detection rates when $L = 32$ bits and SVM has better performance than KNN when $L = 64$, as the training data become larger, the performance of KNN drops down significantly, but the detection rate of SVM is stable.

The VDS achieves high accuracy in detecting known and unknown virus especially in data set1 that the percentage of training set is 25%. But as Figure 4 and 5 shows, their

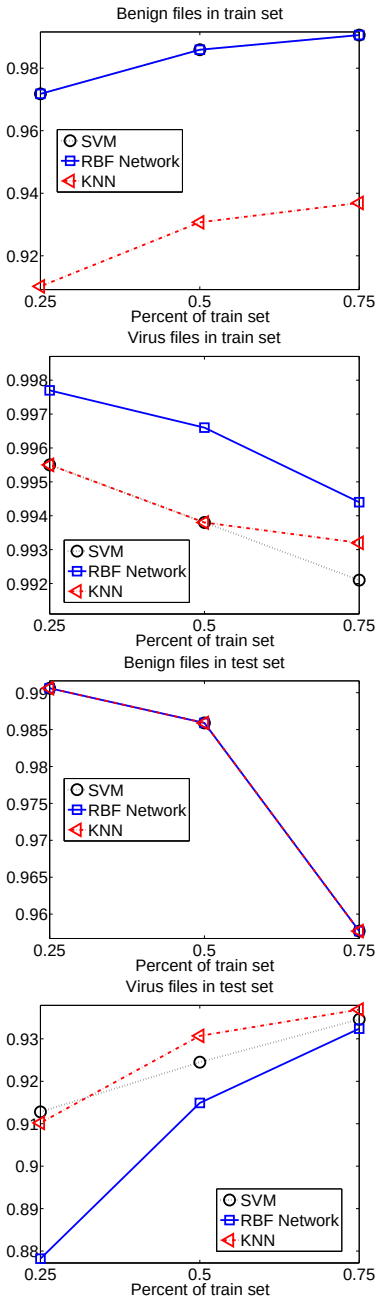

Fig. 4. The average detection rate of SVM, KNN and rbf network when $L = 32$, the files are randomly selected from the dataset

detection rates of benign files in testing set decrease as the number of virus files in training set increases, because some virus files contain amount of benign codes which reduce the thickness of virus information in detector set DT. So additional benign files are used in negative selection to remove benign information in DT. The results also show that the detection rate of virus files in testing set increase as the number of virus files increase in training set. In the dataset, the size of benign files is much larger than that of virus files, it is a general situation in the computer software environment. As the size of files in the dataset grows larger, the decreasing part and increasing part neutralize each other,

the detection rate stays at a stable point. One of our future work is focusing on how to raise this stable point.

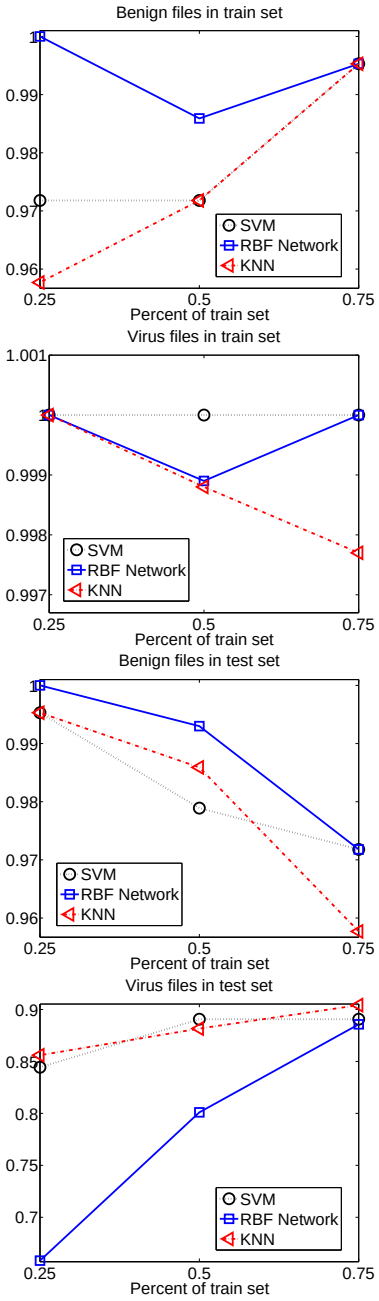

Fig. 5. The average detection rate of SVM, KNN and rbf network when $L = 64$, the files are randomly selected from the dataset

V. CONCLUSIONS

Inspired by the negative selection and clonal selection algorithms in AIS, the VDS is proposed for virus detection in this paper. Experimental results showed that the VDS with the rbf-SVM classifier has a strong generalization ability in detecting unknown virus with low false positive rate. 64-bit detectors have better performance than 32-bit detectors to the virus files in training set, but 32-bit detectors have better performance on virus file in the test set. The discrimination error often happens when the size of a file is too small, because of little information utilized by VDS. As the correlation information between data fragments in these files does not completely used, our future work will focus on introducing a proper correlation value between data fragments into VDS to reduce the false positive rate further.

VI. ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under grant No.60673020 and 60875080, and partially supported by the National High Technology Research and Development Program of China (863 Program), with grant No.2007AA01Z453.

REFERENCES

[1] Kerchen, P., Lo, R., Crossley, J., Elkinbard, G., and Olsson, R. Static Analysis Virus Detection Tools for Unix Systems, pp.4–9. 13th National Computer Security Conference, 1990.

[2] Hofmeyr, S.Forrest, S.and Somayaji, A. (1998) "Intusion Detection Using Sequences of System Calls", Journal of Computer Security, vol. 6, pp.151–180.

[3] Kephart, J.O. and Arnold, W.C.Automatic Extraction of Computer Virus Signatures. 4th Virus Bulletin International Conference, pp. 178–184, 1994

[4] Aickelin, U., Greensmith, J., and Twycross, J.(2004) "Immune System Approaches to Intrusion Detection—a Review". In the Proceeding of the Third International Conference on Artificial Immune Systems (ICARIS–04), pp.316–329.

[5] S.Forrest, A.S.Perelson, L.Allen and R.Cherukuri, "Self-nonself discrimination in a computer", in Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, pp.16–17, Los Alamitos, CA: IEEE Computer Society Press, 1994.

[6] Zhenhe Guo, Zhengkai Liu, Ying Tan, Ling Zhang(2006) "An NN-based Malicious Executables Detection Algorithm Based on Immune Principles", pp.5–6.

[7] Patrik D'haeseleer, S.Forrest, and P.Helman, "An immunological approach to change detection: algorithms, analysis and implications," IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp. 110–119, 1996.

[8] Paul D.Williams, Kevin P.Anchor, John L.Bebo, Gregg H.Gunsch, and Gary D.Lamont, "CDIS: Towards a Computer Immune System for Detecting Network Intrusions",Proceedings 4th International Symposium: Recent Advances in Intrusion Detection (RAID), Davis, CA, USA, pp.117–133, 2001.

[9] J.W.Kappler, N.Roehm, P.Marrack, "T cell tolerance by clonal elimination in the thymus." in Cell, 49:273–280, 1987.

[10] W.E.Paul, Ed., Fundamental Immunology, Raven Press Ltd. New York, pp.88–90, 1989.

[11] Jungwon Kim and Peter J.Bentley,"Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection", pp.19–25. A late-breaking paper, Genetic and Evolutionary Computation Conference (GECCO '99), Orlando, Florida, USA, 1999.

[12] Jungwon Kim and Peter J.Bentley,"An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection,"Proceedings of the Genetic and Evolutionary Computation Conference (GECCO–2001), pp.1330–1337, 2001.

[13] J.O.Kephart,"A biologically inspired immune system for computers", in Proceedings of Artificial Life IV, pp.6–9, MIT Press, Cambridge, MA, 1994

[14] P.D'haeseleer, "A change-detection algorithm inspired by the immune system: Theory, algorithms and techniques", Technical Report CS95–6. The university of New Mexico, Albuquerque, NM, 1995

[15] P.Helman and S.Forrest,"An efficient algorithm for generating random antibody strings", Technical Report CS–94–07, The University of New Mexico, Albuquerque, NM, 1994

[16] V.Vapnik, Estimation of Dependencies Based on Empirical Data. New York: Springer–Verlag, 1992.

[17] H.Drucker, C.J.C.Burges, L.Kauffman, A.Smola, and V.Vapnik,"Support vector regression machines," in Neural Inform. Processing Syst. 9, M.C.Mozer, J.I.Joradn, and T. Petsche, Eds. Cambridge,MA: MIT Press, 1997, pp.155–161.