An efficient Intelligent Cache Replacement Policy Suitable for PACS

Yinyin Wang, Yuwang Yang, and Qingguang Wang

Abstract—**An efficient intelligent cache replacement policy suitable for picture archiving and communication systems (PACS) was proposed in this work. By combining the Support vector machine (SVM) with the classic least recently used (LRU) cache replacement policy, we have created a new intelligent cache replacement policy called SVM-LRU. The SVM-LRU policy is unlike conventional cache replacement policies, which are solely dependent on the intrinsic properties of the cached items. Our PACS-oriented SVM-LRU algorithm identifies the variables that affect file access probabilities by mining medical data. The SVM algorithm is then used to model the future access probabilities of the cached items, thus improving cache performance. Finally, a simulation experiment was performed using the trace-driven simulation method. It was shown that the SVM-LRU cache algorithm significantly improves PACS cache performance when compared to conventional cache replacement policies like LRU, LFU, SIZE and GDS.**

Index Terms—**PACS, cache replacement policy, SVM, hybrid Storage.**

I. Introduction

The PACS is a computer application system dedicated to the process, storage and the transmission of medical images. Through the network, the images information collected by CT, CR, DSA, MRI, gastrointestinal, ultrasound, endoscope and other image equipment are digitized and transferred to the server for classification and storage, so that the relevant image terminals are retrieved as needed, quickly. For instance, the PACS in Yancheng 1st People's Hospital has a comparative 20% annual data update rate each year. The growth of massive data poses numerous challenges to storage systems. Recently, the evolution of storage systems is moving toward high-performance, high-capacity, and low-cost. However, any single storage medium likewise Hard Disk Drive (HDD) and Solid State Drives (SSD) cannot meet the above-mention requirements due to restrictions of their inherent characteristics. Therefore, the hybrid storage media is an efficient solution.

Hybrid storage system composed of storage media with different characteristics. According to the characteristics of data access and the system load conditions, the data request is handed over to the medium best suited to handle, thereby improving the performance of the entire system. The SSD is widely used as a memory in hybrid storage system, with high reliability, low energy consumption and high performance. Considering the high performance of SSD and the low cost of HDD with large capacity, SSD is usually used as the cache of HDD to improve the user experience by providing users with high-speed and large-capacity storage. However, as a cache medium, the SDD is of relatively high cost and limited storage capacity. Therefore, another cache replacement strategy is needed, good enough to manage the hybrid storage.

In PACS, the classic cache replacement strategy has a relatively poor performance, whereas these strategies only consider a certain influence of the cache object (such as size, final access time and frequency). The literature offered machines learning-based approaches to improve the traditional cache replacement strategy likewise Artificial Neural Network (ANN), Support Vector Machine (SVM), and Decision Tree C4.5 (C4.5). However, mention approaches are time-consuming consequently leads to computation overhead besides the prediction accuracy is nasty. More importantly, these policies only focus on the influential factors of cache object itself, ignoring the characteristics of the cache object counterpart entity. Hence, the existing intelligent cache replacement strategy is inefficient in PACS.

A large amount of medical data set produced during the medical treatment. The data set is stored in databases likewise HIS, RIS, and EMR. Many interesting phenomena can be observed via the analysis of these eigenvalues through the mining of data: 1. The complete image flow includes medical examinations, film printing, preliminary reports, final reports, and so on. The image data will be accessed multiple times before the process ends. 2. Image data of patients in the hospital are much more accessed than patients who have been discharged. 3. Image data with positive results are more accessed than negative. 4. Different doctors have quite different ways of accessing image data. 5. Imaging data of the patients who had undergone a surgical operation and who are seriously ill are more likely to be accessed than normal patients. Therefore, this paper hopes to use these eigenvalues to predict the probability that the cache object will be accessed in the future through machine learning. Further, to improve the cache hit ratio through the combination with the classic cache replacement strategy.

Based on the analysis of the cache architecture of PACS and the features of both cache object features and users' habits, this paper combines the SVM algorithm with the classic LRU cache replacement strategy to form the SVM-LRU strategy. This strategy uses a relatively simple SVM algorithm to establish a model that includes influencing factors, features, and users' habits of the cache object. The SVM-LRU strategy has less training time, small computation,

Manuscript received September 24, 2019; revised October 5, 2020.

Yinyin Wang and Yuwang Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China (e-mail: wyywx699@163.com, yuwangyang@njust.edu.cn).

Qingguang Wang is with the Yancheng 1st People's Hospital and Medical School of Nantong University, China (e-mail: wqg699@163.com).

doi: 10.18178/ijmlc.2021.11.3.1043

and high prediction accuracy with improved hit ratio and byte hit ratio performance.

II. Material and Methods

A. Hybrid Storage System Architecture

In PACS, SSD and HDD are commonly used to develop a hybrid storage system [1]-[3]. SSD is used as a cache for HDD. The logical address of the hybrid storage corresponds to the physical address of HDD. In SSD, only a copy of the hot data in HDD is cached. The total capacity of HDD is the entire capacity of the hybrid storage. When there is an access request, the data is first retrieved on the SSD and, if hit it will be read directly from the SSD. If not, the data is transferred from HDD to SSD and read. This architecture is mainly suitable for file-type storage systems. The PACS mainly stores large numbers of image files. Hence, it is very suitable for this architecture.

B. Cache Replacement Strategy

This section reviews cache replacement strategy, which is divided into six types:

1) The strategy based on the Random algorithm (RAND). This type of strategy uses a random number generator. The number belongs to a software or a hardware to determine the replaced object in the main memory. Such, strategy is the simplest and straightforward to implement. However, this kind of strategy does not consider the various factors that affect the object at all, so the hit ratio is relatively low, hence is less practical.
2) Strategy Based on the LRU [4]. This strategy takes objects that have not been accessed for the longest time in the recent past as replaced objects. The advantage is the relatively easy implementation, low time complexity, and has a good performance in applications with unified cache sizes. However, it ignores factors such as the frequency and size of the object, and there is a problem of cache pollution.
3) The strategy based on the Least Frequently Used algorithm (LFU) [5]. This type of strategy selects the object that has been least accessed recently as the object to be replaced, which can help avoid caching pollution problems. However, this policy is very difficult to implement. It sets a counter for each object and selects a fixed clock to count for each counter. Similarly, it only considers the frequency factor of the cache object and ignores other factors.
4) The strategy based on the SIZE [6]. This strategy replaces the object of maximum size from the cache when a new object requests for space. The strategy is simple and easy. However, the cache can be polluted by small objects that are not accessed again but are difficult to replace. This policy is suitable for WEB cache replacement applications, which have a high cache hit ratio but a low byte-hit ratio.
5) The strategy based on function. Likewise, Greedy Dual-Size (GDS) and Greedy Dual-Size Free (GDSF) [7] can optimize cache performance by selecting the appropriate weighted parameters. Multiple influential factors can be considered to handle different application scenarios. However, it is a very difficult task to choose the appropriate weighted parameters for this strategy. Furthermore, new problems may be produced in the calculation of function values.
6) Intelligent cache replacement strategy. Numerous intelligent cache replacement policies have been proposed in recent years. The strategies proposed in References [8]-[12] can generally be divided into two categories. In the first category, intelligent algorithms are used independently as cache replacement policies; in the second category, intelligent algorithms are combined with conventional cache replacement policies. Both of these approaches rely on the prediction of future access probabilities to enhance cache performance.

However, these intelligent cache replacement policies depend only on the intrinsic properties of the cached items alone (e.g., filesize, last access time, and access frequency). In our method, medical data is mined to identify the patients who correspond to the PACS's cached items. A machine learning algorithm is then used to construct a predictive model that is based on the variability of the treatment process. This model is then used to predict the future access probability of each cached item, thus improving cache performance.

C. Support Vector Machine

The support vector machine (SVM) is one of the most robust and accurate methods in all well-known machine learning algorithms. SVM has been used successfully in a wide range of applications such as text classification, Web page classification and bioinformatics applications [13], [14]

Consider the problem of separating the set of training data $(x_1, y_1), (x_2, y_2), \cdots (x_m, y_m)$ into two classes, where $\boldsymbol{x_i} \in \boldsymbol{R}^N$ is a feature vector and $y_i \in \{-1, +1\}$ its class label. If we assume that the two classes can be separated by a hyperplane $\boldsymbol{\omega x} + \boldsymbol{b} = 0$ in some space $\boldsymbol{H}$, and that we have no prior knowledge about the data distribution, then the optimal hyperplane is the one which aximizes themargin [15]. The optimal values for $\boldsymbol{\omega}$ and b can be found by solving a constrained minimization problem, using Lagrange multipliers $\boldsymbol{\alpha_i}(\boldsymbol{i} = 1, \cdots \boldsymbol{m})$.

$$\boldsymbol{f}(\boldsymbol{x}) = sgn(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{K}(x_i, x) + \boldsymbol{b}) \quad (1)$$

where α_i and b are found by using an SVC learning algorithm [15]. Those χ_i with nonzero α_i are the "support vectors". For $\boldsymbol{K}(x, y) = x \cdot y$ this corresponds to constructing an optimal separating hyperplane in the input space $\boldsymbol{R}^N$.

Based on results reported in the literature, in this paper we use the kernel $\boldsymbol{K}(x, y) = exp\{-\gamma \boldsymbol{X}^2(x, y)\}$ for histogram features $\boldsymbol{H}$, and for local features we use the kernel $\boldsymbol{K_L}(\boldsymbol{L_h}, \boldsymbol{L_k}) = \frac{1}{2}\left\lfloor \boldsymbol{K}(\boldsymbol{L_h}, \boldsymbol{L_k}) + \boldsymbol{K}(\boldsymbol{L_k}, \boldsymbol{L_h}) \right\rfloor$, with

$$\boldsymbol{K}(\boldsymbol{L_h}, \boldsymbol{L_k}) = \frac{1}{\boldsymbol{n_h}} \sum_{j_h=1}^{n_h} \max_{j_k=1,\cdots n_k} \{\boldsymbol{K_l}(\boldsymbol{l_{jh}}, \boldsymbol{l_{jk}})\} \quad (2)$$

where $\boldsymbol{L_i} = \{\boldsymbol{l_{ji}}\}_{j=1}^{n_i}$ and $\boldsymbol{l_{ji}}$ s a jet descriptor of interest point $\boldsymbol{j}$

in sequence i and

$$K_l(x,y) = exp\left\{-\rho(1-\frac{\langle x - \rangle}{\|x-\mu_x\|\cdot\|y-\mu_y\|})\right\} \quad (3)$$

where μ_x is the mean of x .

III. SVM-LRU Intelligent Cache Replacement Strategy for PACS

The SVM-LRU intelligent cache replacement strategy for PACS uses SVM algorithm to predict the probability that the cache object will be accessed in the future by considering the eigenvalues of the patient corresponding to the cache object in the medical process to, through the combination with LRU, improve PACS cache replacement performance.

A. SVM-LRU Intelligent Cache Replacement Strategy Framework

In his section, the SVM-LRU intelligent cache replacement policy can be presented and framework is shown in Fig. 1. The framework consists of three functions: Extract-Transform-Load (ETL) component, offline component, and an online component.

The ETL component is responsible for extracting data in heterogeneous data sources (such as HIS, RIS, EMR, etc.) distributed in hospitals into the temporary intermediate layer for cleaning, conversion, integration, and finally loading into the target database, which lays the basis for later data analysis and data mining [16], [17].

The offline component does not handle users' access requests directly. It trains SVM-LRU policies when the server is idle. The updated SVM-LRU policy is then applied to the cache manager for online components.

The online component manages the cache manager. In case of an access request from users, first, the eigenvalue of the cache object will be gained from the target database, which would then be put into cache manager use, followed by the fulfilment of cache management by using SVM-LRU policy.

B. SVM-LRU Intelligent Cache Replacement Strategy

LRU policy is a classical cache replacement policy. However, the LRU policy is subject to cache pollution, which means that unwanted objects reside on the cache for a long time. In other words, a new object will be inserted in the LRU, at the top of the cache stack. If the object is no longer accessed in the future, it takes a long time to move down to the bottom of the stack before removing it from the cache.

SVM is combined with LRU to form a new SVM-LRU to reduce cache pollution in LRU. The SVM-LRU workflow is as follows. When the users' requests object X, SVM predicts P the probability that the object will be accessed in the future. If P is greater than or equal to the threshold ξ, object X is determined as hot data, which will be placed at the top of the cache stack. Otherwise, object X is cold, which will be placed in the middle of the cache stack for a quick elimination:

The SVM-LRU strategy pseudo code is as follows.

```
Algorithm: SVM-LRU
Input: each file X requested by user
If X in SSD
    Cache hit occurs return X //file X read completion
Else
    Fetch X from HDD to SDD
End
While no enough space in SDD for X
    Evict Y such that Y in the bottom of the cache stack
    Insert X in the bottom of the cache stack
    Cache hit occurs  return X //file X read completion
End
Use SVM-LRU algorithm update the probability $P_X$ of X
Predict whether X is cold data or hot data
If X is cold data
    Move X to the middle of the cache stack
Else
    Move X to the top of the cache stack
End
Output: X
```

IV. Effect Assessment and Discussion

A. Cache Replacement Strategy Evaluation Index

There are two main indicators used to evaluate cache storage performance, Hit Ratio (HR) and Byte Hit Ratio (BHR). The hit ratio represents the percentage of access requests obtained from the cache among total access requests. And the byte hit ratio represents the percentage of bytes obtained from the cache among the total bytes of access requests.

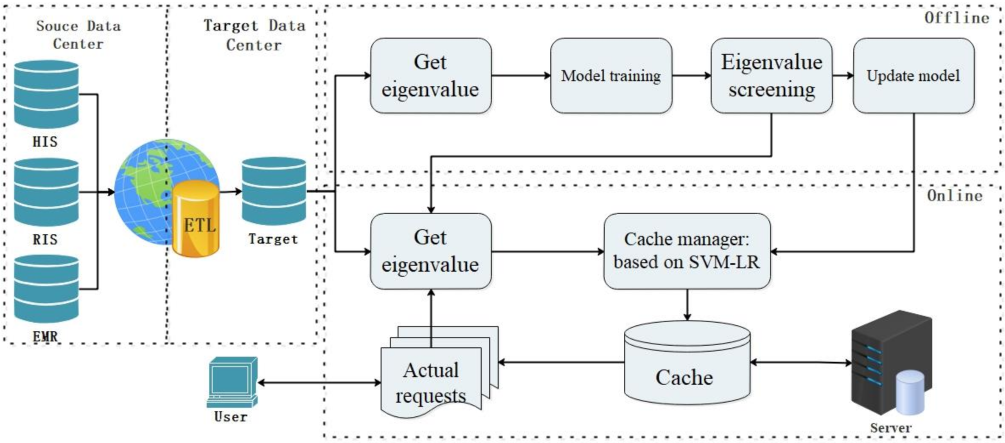

Fig. 1. SVM-LRU intelligent cache replacement strategy framework.

HR and BHR calculation methods are as the following Formulas. Each request corresponds $i(1 \le i \le N)$ to a cached object. m_i represents the size of the object i . q_i indicates object i hit or not, if hit q_i=1 , not hit q_i=0 .

$$HR=\frac{\sum_{i=1}^{N} q_i}{N} \times 100\% \tag{4}$$

$$BHR=\frac{\sum_{i=1}^{N} q_i m_i}{\sum_{i=1}^{N} m_i} \times 100\% \tag{5}$$

HR and BHR have different focuses. HR focuses on reducing users' response time and achieving a better user experience. While BHR focuses on reducing the erase frequency of SSD and extending its service life. It is very difficult for a cache replacement strategy to make HR and BHR both get optimal performance [18]. This is because the strategy of improving HR usually favour small-sized objects instead of large-sized objects, thus reducing BHR. In contrast, strategies that do not favour small-sized objects tend to increase BHR at the expense of HR

B. Comparison between SVM-LRU Strategy and Classic Strategy

The Track Driven Simulation method is used to conduct experiments [19] he experimental data was derived from the PACS of Yancheng No. 1 People's Hospital, and the data was collected between 7th May 2018 and 27th May 2018 (a total of 21 days). Before the experiment, the stop point of the experiment, namely the infinite cache size, should be determined. Infinite cache means that there is enough space to store all cached objects without replacing any cached objects. In addition, the capacity of infinite cache is the total sum of all cache object sizes, among which HR and BHR can reach maximum values. Considering the cost element, however, infinite cache is impossible to realize. The infinite cache obtained through the experiment is 16134GB. $HR_{\max} = 42.36\%$ $BHR_{\max} = 45.18\%$. So 9 levels are selected, and the cache capacity is from 2^5 GB to 2^{13} GB.

The SVM-LRU strategy and the classic GDS, LRU, LFU, and SIZE were used for experiments. Fig. 3 and Fig. 4 show HR and BHR in different strategies with different cache capacities. As the cache capacity increases, HR and BHR in all strategies increase. When the cache capacity is close to the capacity of the infinite cache, HR and BHR become stable and close to its maximum value.

As shown in Fig. 3, it is obvious that SVM-LRU has improved the hit ratio of the classic LRU strategy as regards to HR, indicating ll at the introduction of the SVM in the LRU strategy is effective. Since the frequency at which the image cache stored in PACS is accessed is not much different under the same circumstances, that is, there are fewer over-heated or over-cooled cache files, so the LFU strategy that introduces frequency factors has the worst performance. Similarly, the large difference in the cache size of the image stored by PACS, for example, the cache files such as ultrasonography, DR, etc. are only1~10MB, while the cache file size of CT, MIR and dimensions. When the cache capacity is 32GB, others have hundreds of MB. So when the introduced dimension factors like SIZE, GDS are smaller in cache capacity, it has more advantages over other strategies. But with the increase in cache capacity and the over-abandonment of large files, HR's growth rate is significantly slower than other strategies. In GDS, there is not much difference between and SVM-LRU and HR. GDSF has more advantages with small cache capacity, otherwise, SVM-LRU has more advantages.

From the point of BHR, as shown in Fig. 4, size factors are not considered in SVM-LRU, LRU, and LFU, so the trend of BHR is basically the same as that of HR. Small-size objects are preferred by SIZE and GDS, so a high HR is obtained at the expense of low BHR. When the cache capacity is small, HR is superior to SVM-LRU, but BHR is significantly inferior to it.

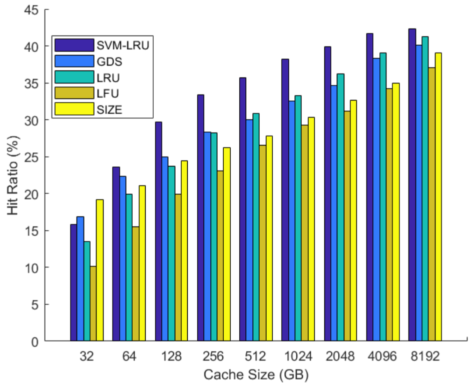

Fig. 3. Comparison of HR for multiple algorithm.

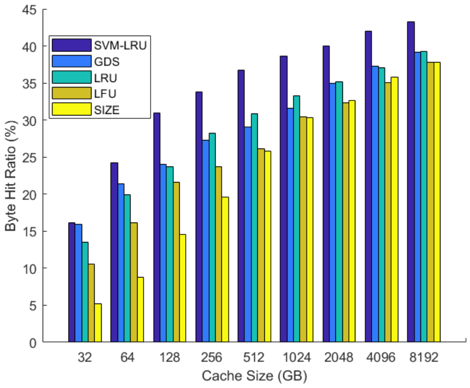

Fig. 4. Comparison of BHR for multiple algorithms.

To better illustrate the advantages of SVM-LRU strategy, the Improvement Ratios (IR) of HR and BHR performance are introduced to evaluate the advantages and disadvantages of different strategies. IR calculation formula is as formula (16), where PM represents the suggested Proposed Method (PM) and CM represents a comparative model (CM). Table I summarizes IR statistics of SVM-LRU and other strategies.

$$IR=\frac{(PM\text{-}CM)}{CM} \times 100\% \tag{6}$$

Form the overall, the results in Table I show that BHR in SVM-LRU is superior to other policies in all cache capacity. But HR is only inferior to SIZE, GDS and GDSF when cache capacity is relatively small, for such strategy often sacrifice BHR for higher HR, especially SIZE, which only focuses on $IR_{HR} = -17.06\%$, $IR_{BHR} = 209.79\%$. It can be seen that

SVM-LRU has got a bigger BHR advantage at a relatively small loss of HR.

TABLE I: IR STATISTICS FOR SVM-LRU OVER CLASSICAL STRATEGIES

Cache size (GB)	SVM-LRU Over GDS(%)		SVM-LRU Over LRU(%)		SVM-LRU Over LFU(%)		SVM-LRU Over SIZE(%)	
	HR	BHR	HR	BHR	HR	BHR	HR	BHR
32	-6.39	1.45	17.00	19.29	55.96	52.99	-17.60	209.79
64	5.45	13.48	18.48	21.80	51.90	50.34	11.64	175.57
128	18.93	29.01	25.14	30.32	49.20	43.22	21.45	111.84
256	17.97	23.83	18.27	19.75	44.57	42.56	27.62	72.96
512	18.99	26.53	15.56	18.80	34.49	40.58	28.58	42.44
1024	17.26	22.46	14.82	16.24	30.47	27.13	25.83	27.38
2048	15.11	14.52	10.17	13.70	27.89	23.81	21.94	22.37
4096	9.01	12.71	6.92	13.47	22.12	19.85	19.22	17.37
8192	5.38	10.63	2.57	10.35	14.17	14.70	8.41	14.49

Compared to LRU, SVM-LRU significantly improves HR and BHR performance and has the best performance when the cache capacity is 128GB, where $IR_{HR} = 25.14\%$, $IR_{BHR} = 30.32\%$. SVM-LRU can predict and replace cold data in the cache faster, and it can significantly improve cache performance especially when the cache capacity is relatively small, which indicates that the introduction of the SVM into LRU is correct and effective. SVM-LRU has a good performance on HR and BHR compared to LFU, indicating that access frequency is not a key factor of the hybrid storage cache replacement strategy like PACS. Comparing SVM-LRU with GDS and SVM-LRU with SIZE, HR is slightly worse only when the cache capacity is 32GB. HR and BHR in other cases have been significantly improved, which indicates that the strategy of focusing on dimensions is not suitable for PACS hybrid storage cache.

In summary, SVM-LRU is better than other classical strategies, and its BHR is good enough. HR has a good performance when the cache capacity is large enough, and it is generally superior to other classical strategies.

V. CONCLUSION AND FUTURE WORKS

This paper proposes an intelligent cache substitution strategy for PACS, which combines the SVM with the LRU to form the SVM-LRU intelligent cache replacement strategy. Different from the existing cache replacement policy, only the nature of the cache object itself is used to formulate the strategy (such as size, final access time and frequency). The SVM-LRU strategy oriented towards PACS obtains the eigenvalues in the medical process that belongs to the patients corresponding to the cached object through mining the medical data. Experimental results show that SVM-LRU significantly improves cache performance compared to LRU, LFU, SIZE and GDS.

The SVM-LRU strategy proposed in this article is based on the example of PACS in hospitals. Its core ideas can be extended to industries such as electricity, banking, aerospace, and telecommunications. Which are similar to hospital systems with a large amount of data, a large number of data dimensions, and strong data relevance. The eigenvalues are obtained through data mining. The machine learning algorithm is used to establish a model to predict the probability that the cache object will be accessed in the future, thus improving the cache performance.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yinyin Wang performed the data analyses and wrote the manuscript. Yuwang Yang provided the conception of the study. Qingguang Wang contributed significantly to analysis and manuscript preparation.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 31471415, Grant 61671244, and Grant 61773206, in part by the National Defense Technology Foundation Research Project under Grant JCKY2018****303 and Grant JCKY2018****001, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011204, in part by the Key Research and Development project in Jiangsu under Grant SBE2018310371, in part by the Scientific Research Foundation of Nanjing University of Posts and Telecommunications under Grant NY217148, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20170131, in part by the Key Technology and General Program of Jiangsu Province under Grant BE2018393, and in part by the Key Industrial Technology Innovation Project of Suzhou City under Grant SYG201826.

REFERENCES

[1] R. Micheloni, L. Crippa, and M. Picca, "Hybrid storage," *Inside Solid State Drives*, Springer, pp. 61-77, 2013.

[2] R. Hemmati, M. Shafie-Khah, and J. P. S. Catalao, "Three-level hybrid energy storage planning under uncertainty," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 2174-2184, 2018.

[3] Z. Chen, W. J. Liu, X. Zhang, and H. L. Bo, "Review on HDD-SSD hybrid storage," *Journal of Computer Applications*, vol. 37, pp. 1217-1222, 2017.

[4] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, vol. 28, no. 2, pp. 202-208, 1985.

[5] D. Lee, J. Choi, J. H. Kim et al., "On the existence of a spectrum of policies that subsumes the least recently used (LRU) and least frequently used (LFU) policie," in *Proc. ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 1999, pp. 134-143.

[6] G. P. Sajeev and M. P. Sebastian, "A novel content classification scheme for web caches," *Evolving Systems*, vol. 2, no. 2, pp. 101-118, 2011.

[7] J. B. Patil and B. V. Pawar, "GDSF#, a better caching algorithm that optimizes both hit rate and byte hit rate in web proxy servers,"

International Journal of Computer Science & Applications, vol. 5, no. 4, pp. 1-10, 2008.

[8] X. Y. Huang and Y. Q. Zhong, "Web cache replacement algorithm based on Multi-Markov chains prediction model," *Microellectronics & Computer*, vol. 5, pp. 123-125, 2014.

[9] K. Ma, B. Yang, Z. Yang *et al.*, "Segment access-aware dynamic semantic cache in cloud computing environment," *Journal of Parallel & Distributed Computing*, vol. 110, pp. 42-51, 2017.

[10] Q. Yang, H. H. Zhang, and T. Li, "Mining web logs for prediction models in WWW caching and prefetching," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 473-478.

[11] J. H. Chang and W. S. Lee, "estWin: Adaptively monitoring the recent change of frequent itemsets over online data streams," in *Proc. Cikm*, 2003, pp. 536-539.

[12] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "Intelligent Web proxy caching approaches based on machine learning techniques," *Decision Support Systems*, vol. 53, no. 3, pp. 565-579, 2012.

[13] O. Nasraoui, "Web data mining: Exploring hyperlinks, contents, and usage data," *ACM SIGKDD Explorations Newsletter*, vol. 10, pp. 23-25, 2008.

[14] N. Markatchev and C. Williamson, "WebTraff: A GUI for Web proxy cache workload modeling and analysis," in *Proc. the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, IEEE Computer Society, 2002, p. 356.

[15] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988-999, 1999.

[16] Q. Yao *et al.*, "Design and development of a medical big data processing system based on Hadoop," *Journal of Medical Systems*, vol. 39, no. 3, pp. 23, 2015.

[17] G. Hughes, "Tjur's R^2 for logistic regression models is the same as Youden's index for a 2×2 diagnostic table," *Annals of Epidemiology*, vol. 27, no. 12, pp. 801-802, 2017.

[18] S. Romano and H. Elaarag, "A neural network proxy cache replacement strategy and its implementation in the Squid proxy server," *Neural Computing & Applications*, vol. 20, no. 1, pp. 59-78, 2011.

[19] R. A. Uhlig and T. N. Mudge, "Trace-driven memory simulation: A survey," *ACM Computing Surveys*, vol. 29, no. 2, pp. 128-170, 1997.

Yinyin Wang was born in Yancheng, Jiangsu, China. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include machine learning, high performance computing and medical big data.

Yuwang Yang received the B.S. degree from North Western Polytechnical University in 1988, the M.S. from the University of Science and Technology of China in 1991, and the Ph.D. degree from the Nanjing University of Science and Technology in 1996. He is currently a professor with the School of Computer Science and Engineering, NUST. His research interests are high performance computing, machine learning, and intelligent system.

Qingguang Wang received the B.S. degree from Peking University, and he is now the chief pharmacist of Yancheng 1st People's Hospital and an associate professor of Medical School of Nantong University.