

Neural Network based Mobility aware Prefetch Caching and Replacement Strategies in Mobile Environment

Hariram Chavan
Information Technology,TEC,
Mumbai University, India
chavan.hari@gmail.com

Suneeta Sane
Computer Technology,
V. J. T. I., Mumbai, India
sssane@vjti.org.in

H. B. Kekre
MPS of Tech. Mgmt & Engg
NMIMS University, India
hb_kekre@nmims.edu

Abstract—The Location Based Services (LBS) have ushered the way mobile applications access and manage Mobile Database System (MDS). Caching frequently accessed data into the mobile database environment, is an effective technique to improve the MDS performance. The cache size limitation enforces an optimized cache replacement algorithm to find a suitable subset of items for eviction from the cache. In wireless environment mobile clients move freely from one location to another and their access pattern exhibits temporal-spatial locality. To ensure efficient cache utilization, it is important to consider the movement direction, current and future location, cache invalidation and optimized prefetching for mobile clients when performing cache replacement. This paper proposes a Neural Network based Mobility aware Prefetch Caching and Replacement policy (NNMPCR) in Mobile Environment to manage LBS data. The NNMPCR policy employs a neural network prediction system that is able to capture some of the spatial patterns exhibited by users moving in a wireless environment. It is used to predict the future behavior of the mobile client. A cache-miss-initiated prefetch is used to reduce future misses and valid scope invalidation technique for cache invalidation. This makes the policy adaptive to clients movement behavior and optimizes the performance compared to earlier policies.

Keywords—Location Based Services, Caching, backpropagation, cache-miss-initiated prefetch, cache replacement policy.

I. INTRODUCTION

The fast development of wireless communication system and advancement in computer hardware technology has led to the seamlessly converged research area called mobile computing. The mobile computing environment enables unrestricted mobility of the mobile user. The notion of anything, anytime, anywhere has been brought by mobile devices such as laptops, PDAs and cell phones. Mobility and portability have created an entire new class of applications in mobile database [1][2].

In last few years, the new fertile area for researchers in mobile computing is LBS. The LBS have ushered the way mobile applications access and manage Mobile Database System. Location based services are based on the location of mobile clients. The different value-added applications which specifically target the mobile clients location as context information are traffic condition, tourist information system, weather information system, emergency system, location-dependent advertising and disaster management systems [3][4][5][7][11][14].

The location dependent advertisement messages can be delivered to users who are within a specific area such as Departmental Store. The complete weather forecasts and weather alert notifications can be delivered to users who are in the area where weather conditions will change. If a user is under circumstances of an emergency, and the exact location of the user can be obtained then other people or organizations can offer help to the user more quickly, easily and efficiently [6][14][26].

LBS, being wireless in nature are plagued by mobility constraints like limited bandwidth, client power and intermittent connectivity. It is observed that prefetching and data caching at mobile client helps to address some of these challenges and acts as an effective antidote for the listed limitations [1][2][3][4][7][11]. In general, there are three important issues involved in the client cache management [13][19]:

- 1) A cache replacement policy finds a suitable subset of items for eviction from the cache when there is no adequate space to accommodate a new data item.
- 2) A cache prefetching policy finds suitable subsets of data items in which user maybe interested in future; and
- 3) A cache invalidation scheme maintains data consistency between the client cache and the server.

Prefetching improves the system performance in mobile environments but consumes system resources, such as bandwidth and power. A simple consequence of a cache miss is a user based reactive information query and thus slower data access [27]. So when a cache miss happens, instead of sending an uplink request only for the cache-missed data item, the client requests several associated data items to reduce future cache misses. The most important aspect is, the client can prefetch more than one associated data items by an uplink request with a little additional cost. Thus, prefetching several data items in one uplink request can save additional uplink requests [14][15][16]. In this paper, we have optimized the issue of prefetching by implementing cache-miss-initiated prefetching.

A cache invalidation scheme maintains data consistency between the client cache and the server. There are two kinds of cache invalidation methods for mobile databases: temporal-dependent invalidation caused by data updates and location-dependent invalidation caused by client mobility. The

temporal-dependent invalidation is handled by Invalidation Report (IR) schemes which are variations of the basic IR approach. In location-dependent services, a previously cached data value may become invalid when the client moves to a new location. The valid scope of an item is defined in the region within which the item is valid (i.e scope invalidation scheme) [17][18][19].

In this paper, we propose a neural network based mobility aware cache replacement policy that takes into account both the temporal and spatial locality of clients access pattern. The proposed policy is called Neural Network based Mobility aware Prefetch Caching and Replacement Policy (NNMPCR). We validate with simulation (NS2) results that mobile clients using NNMPCR are able to achieve significant improvement in cache hit ratio compared to clients using other existing cache replacement policies.

This paper is organized as follows. Section II provides a survey of existing cache replacement policies. Section III describes the Neural Network as a Location Predictor: preliminaries. In Section IV, the data prefetching concept is described. In section V, we propose a new cache replacement strategy, NNMPCR for mobile clients. Simulation model of NNMPCR is presented in section VI. The results are presented in Section VII. Section VIII concludes the paper.

II. RELATED WORK

Through extensive literature survey, it is realized that the issue of data caching in the mobile environment was first addressed in [1][2] by D. Barbara et.al. Since then caching in the mobile environment has been addressed by many researchers. Most of the existing cache replacement policies use cost functions to incorporate different factors including access frequency, update rate, size of objects, temporal score, spatial score etc. [3][4][5][6][10][18].

The traditional temporal-dependent Least Recently Used (LRU) policy has been studied widely in the literature. LRU drop a page from buffer based on last reference. This policy suffers from decision making with very limited data and has been modified by O'Neil et.al [8]. The LRU-K [8] discriminates between frequent and infrequent pages based on the built-in notion of aging and is able to cope-up with evolving access patterns. The LRU-K policy has been designed by considering the history of the last k references (where $k \geq 2$). However, in LBS, the replacement policy must consider temporal as well as spatial dependency of data

Furthest Away Replacement (FAR) [9] is a location-aware cache replacement policy. FAR selects replacement victim based on the current location and movement direction of mobile client. The data items which are furthest from the mobile client will be evicted first. The assumption is that they wont be visited in the near future. FAR does not take into account access probability, valid scope area and random mobility of mobile client.

Mobility-Aware Replacement Scheme (MARS) [6] is a gain-based cache replacement policy for the mobile environment. The cost function of MARS considers temporal and spatial score together while making cache replacement decisions. Access probabilities, update and query rate results into

temporal score. The spatial score consists of client location, the location of data objects and client movement direction. For LBS, spatial score dominates and must consider the impact of clients anticipated location while deciding cache replacement which still remains unexplored.

Probability Area Inverse Distance (PAID) proposed by Baihua Zheng et. al [18] is based on Cache-Efficiency Based scheme (CEB). CEB is used for balancing the overhead and the precision of performance criterion. PAID, undertakes the valid scope area, data distance and access probability for cache replacement. The valid scope of a data value is defined as the geographical area within which the data value is valid and has been used for cache invalidation. PAID neither takes into account the data size nor data updates while cache replacement. The CEB concept is based on inscribe circle within polygon to represent the valid scope area. If the polygon is thin, then the inscribed circle covers less valid scope area. This problem is analyzed, and a modified CEB with a median circle is proposed by Kahkashan Tabassum et. al [10]. The median circle radius is in between the inscribe circles radius and the outer circle radius.

Ajey Kumar, Manoj Misra and A. K. Sarje [3][4][5] proposed Predicted Region based Cache Replacement Policy (PPRRP) and Weighted Predicted Region based Cache Replacement Policy (WPPRRP) policies. The cost function of these policies considers access probability, weighted data distance from predicted region, valid scope area and data size in cache. Predicting future user influence region and assigning priority to the data items in the current vicinity of mobile client helps to increase the cache hit ratio.

III. NEURAL NETWORK AS A LOCATION PREDICTOR: PRELIMINARIES

The mathematical model of an artificial neural network (ANN) emulates the functioning of a biological neural system. The architecture of feed forward network is shown in Fig.1. It does not have feedback connections, but errors are back-propagated during supervised training as shown in Fig.2.

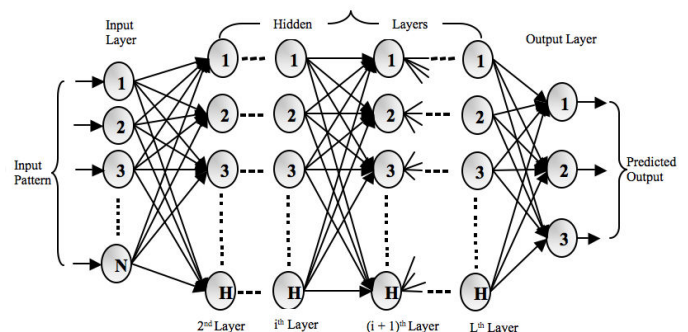


Fig. 1. Multilayer feedforward network architecture

The input layer neurons are linear, whereas neurons in the hidden and output layers have sigmoidal activation functions. For input neurons

$$f(x) = x \quad (1)$$

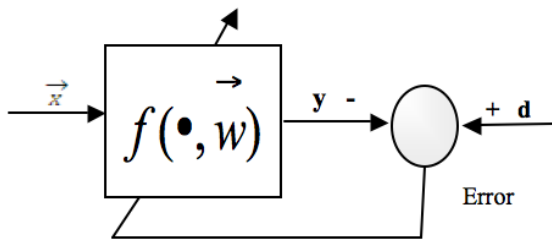


Fig. 2. Neural network training model with error backpropagation

For sigmoidal neurons in the hidden and output layers,

$$f(x) = (1 + e^{-x})^{-1} \quad (2)$$

Assuming we have N training vector pairs (input, desired)

$$T_p = \{X_k, D_k\}_{k=1}^N \quad (3)$$

Where X_k is the k^{th} input pattern and D_k is the k^{th} desired vector response when X_k^{th} input is applied to the network. The gradient of the pattern error is used to reduce the global error over the entire training set. For k^{th} training pair (X_k, D_k) the error is

$$E_k = D_k - f(y_k) \quad (4)$$

For a neuron j with activation function $f(x)$, the delta learning rule for j 's i^{th} weight w_{ji} is given by:

$$\Delta w_{ji} = \eta(d_j - y_j)f'(v_j)x_i \quad (5)$$

Where,

η	Is a small constant called learning rate
$f(x)$	Is the neuron activation function (sigmoid)
d_j	Is the desired output
y_j	Is the actual output
v_j	Is the weighted sum of the neurons inputs
x_i	Is the i^{th} input

The v_j is given by

$$v_j = \sum x_i w_{ji} \text{ and } y_j = f(v_j) \quad (6)$$

The error gradient for each pattern is computed and used to change the weights between layers of network. Adjusting the weights between the pairs of layers and recalculating the outputs is an iterative process and carried out until the errors fall below a tolerance level. Learning rate parameters scale the adjustments to weights. Massive parallelism, learning, adaptivity and fault tolerance are the desirable characteristics of ANN. Researchers from many scientific disciplines are designing ANNs to solve a variety of problems and location prediction is one of them [20].

Intuitively location (mobility) prediction uses the historical movement patterns of mobile client to determine his/her possible future locations [21][22][23]. Knowing in advance where a mobile client is heading allows us to take proactive measures. The motion of mobile client within the mobile network and

<p>The back-propagation algorithm [24][29]:</p> <ol style="list-style-type: none"> 1. Initialize the weights to small random values. 2. Choose a encoded distance and direction pattern X_k from the training set T_p and apply it to the input layer. 3. Propagate the distance and direction signal forwards through the network. 4. Compute the difference between anticipated encoded location and actual output at the output layer (δ error). 5. Use the error calculated in step (4) to compute and update weight change between pairs of layers. 6. Update all weights of the network in accordance with the changes computed in step (5). Hidden to output layer weights $w_{jh}(k+1) = w_{jh}(k) + \Delta w_{jh}(k)$ (7) Input to hidden layer weights $w_{hi} = w_{hi} + \Delta w_{hi}(k)$ (8) Where $w_{jh}(k)$ and $\Delta w_{hi}(k)$ are weight changes computed in step (5). 7. Repeat step (2) through (6) until the global error falls below a predefined threshold (0.001).
--

Fig. 3. Backpropagation algorithm.

successive list of connections are the better alternative to prune the data entries. This pruning ensures that the location of mobile client always represents one of the possible access points in the network.

In typical mobile networks, mobile client exhibits some degree of regularity in the mobility pattern. By exploring these regularities in mobility pattern, we can predict the future location of mobile client. Since privacy is an issue that may arise when tracking of the mobile clients, it is better to use generalized movement patterns. The generalized patterns better handle local as well as irregular movement patterns. In this paper, such generalized historical movement patterns are used to train the neural network using backpropagation algorithm (Fig. 3) to predict the future location of mobile client [22][23].

IV. DATA PREFETCHING

Prefetching is concerned with improving the system performance. In mobile database system, prefetching is used as an extension to caching to improve the performance. In LBS, prefetching is the only way to avoid the need of refreshing with location change [15][27]. It is an attractive solution to reduce access latencies perceived by mobile users. Prefetching implies the prediction of information needed by the user in future. In order to effectively limit and filter potential prefetched information into the most likely future location context, one should consider users, moving speed, moving direction, habits, preferences, interests and available bandwidth [14][15][16]. In this paper, prefetching is extended with a cache-miss-initiated prefetch (CMIP) [14] scheme. As prefetching improves MDS performance with intelligent data caching and good replacement policy, this paper proposes an ANN based location prediction, CMIP based prefetching and valid scope invalidation based complete NNMPCR policy in the mobile environment.

Caching frequently accessed data on the client improves MDS performance. If cache miss, the client has to send an

expensive uplink request to fetch the queried data item. To minimize expensive uplink bandwidth prefetching may be used frequently [14]. As prefetching consumes system resources, a good prefetching scheme for LBS and mobile environment must accomplish [15]:

- Data items related to the mobile clients current location must be prefetched first.
- The movement direction and speed of mobile client must be taken into account.
- Association between data items must be taken into account.
- A good cache invalidation scheme to create free space from non-relevant data items.
- Prefetch only relevant information to save uplink bandwidth.

From above requirements, predicting and prefetching right data items in which mobile clients are interested in the future is important. The CMIP scheme satisfies this requirement by prefetching the highly associated data items. Association rule mining is well researched techniques of data mining introduced by Rakesh Agrawal [30]. It aims to extract frequent patterns, associations among sets of data items in the transaction databases and widely used in telecommunication networks. It finds association rules that satisfy the predefined minimum support and confidence [14][30]. Minimum support is used to find the frequent itemsets and minimum confidence is used to generate association rules from these frequent itemsets.

The user habits, preferences and interest results into a subset of data items which are frequently accessed. This frequently accessed subset of data items fulfills the first requirement of association rule. It has been observed that the set of data items requested over a period of time is related to each other and satisfies the second requirement of association rule [30]. In CMIP scheme, the miss-prefetch set is created with closely related subset of data items using association rules. So when cache misses, many highly associated data items from the miss-prefetch set will be fetched with missed data item. This will reduce future cache misses as well as uplink bandwidth [14].

In NNMPCR, the mobile clients access pattern base frequent itemset and association rule algorithms of [14] are implemented to generate the always-prefetch sets and the miss-prefetch sets. The mobile clients access patterns may change from time to time. Change in access patterns may change the association rules and hence prefetch sets. To adapt this change, NNMPCR periodically re-mines the association rules and prefetch sets.

V. CACHE REPLACEMENT POLICY

LBS is spatial in nature. A data item may show different values if it is queried by mobile clients from different locations. Data distance and valid scope area from the mobile clients current location are important parameters for cache replacement. Larger the distance of data item from the clients current position the probability is low that the client will enter into the valid scope area in the near future. Thus, it is better to replace the farthest data value when cache replacement takes

place. If client movement is random, then it is not always necessary that the client will continue to move in the same direction. Therefore replacing data values which are in the opposite direction of client movement but close to the current position of client may degrade the overall performance [11]. In NNMPCR, neural network predict the future location of mobile client based on historical movement patterns, so there is very less probability that the data item with higher access probability related to previous location will get replaced by new data item.

The ANN used for prediction consists of three layers: input, hidden and output. At input layer, we present the input vector. Thus, the number of neurons in this layer is the same as the number of entries in the input vector. The number of input neurons has a great impact on dimensionality and so encoded activation values (distance and direction) are applied at input layer. The encoding of output corresponds to the encoding of the input. The number of neurons in the hidden layer must be chosen suitably to avoid the under-learning or overfitting [21][22] [23]. So to find the number of neurons in hidden layers one can use empirically-derived rules-of-thumb. The commonly used rules-of thumb are:

- The optimal size of the hidden layer is usually between the size of the input and size of the output layers.
- The number of neurons in hidden layer is the mean of the neurons in the input and output layers.
- Set to something near (inputs + outputs) * 2/3 or never larger than twice the size of the input layer.

The neural network of NNMPCR is trained with nine hidden layer neurons.

The direction will be one of eight N, NE, E, SE, S, SW, W, NW directions and computed as shown in Fig-4. The distance traversed by the mobile client for computed direction is calculated by Euclidian distance formula.

The network is trained using backpropagation learning algorithm (Fig-3) in two passes. Execution of the training equations is based on iterative processes and thus is easily implementable. A pair of patterns is presented (X_k, D_k), where

```
Code for future direction of mobile client
int get_direction (double cur_x,double cur_y,
double last_x,double last_y)
{
    if(cur_x == last_x && cur_y == last_y)
        return 0;
    if(cur_x == last_x && cur_y > last_y)
        return NORTH; // north
        .
        .
        .
    if(cur_x <= last_x && cur_y >= last_y)
        return NORTHWEST; // northwest
    if(cur_x <= last_x && cur_y <= last_y)
        return SOUTHWEST; // southwest
}
```

Fig. 4. Code for the future direction of mobile client

X_k is the input pattern (location and direction) and D_k is the desired pattern (distance). The input X_k causes output response at each neuron in each layer and hence an actual output Ok at the output layer. At the output layer, the difference between the actual and the desired outputs yields an error signal (delta- δ). This error signal depends on the values of the weights of the neurons in each layer. So far, the calculations were computed forward (forward pass). Thus, the forward pass is used to evaluate the output of the neural network for the given input in the existing weights. Now, the algorithm reverts one layer and recalculates the weights of the output layer (the weights between the hidden layer and the output layer) so that the output error is minimized. The algorithm continues calculating the error and computing new weight values, moving layer by layer backward, towards the input (backward or reverse pass). Thus, in the reverse pass, the difference in the neural network output (actual output) with the desired output is compared and fed back to the neural network as an error signal to change the weights of the neural network.

The location of the mobile client is primarily thought of as their geographic coordinates. However connections on the move for a mobile client may be very large. So it is better to consider the movement direction of mobile client through the network and the cell address where the calls were made. This limits to the chances of intermittent contact during the movement. This means, the mobile clients location is always one of a finite set of locations representing one of the possible access points in the network [11][23]. It will significantly reduce the state space. In this paper, we consider such pruned data for further analysis.

The MDS is developed with the following data attributes: valid scope, access probability, data size and distance related to LBS. These databases are percolated to MSS in order to make sure that the data relevant in the valid scope is available in the nearest MSS. The access probability is updated in the database with each access to the cached data item. When cache has no enough space to store queried data item then space is created by replacing existing data items from the cache based on neural network and location proximity model. In the worst case, if all locations are in the proximity then replacement will be based on minimum access probability (P_{ai}), maximum distance (d_{si}) and scope invalidation (v_{si}). If data items have same valid scope, then replacement decision is based on minimum access probability. If valid scope and access probability are equal, then replacement decision is based on maximum data distance. In some cases, data size plays an important role in replacement. If fetched data item size is large enough and requires replacing more than three data items from cache then replacement is based on maximum equivalent size with minimum access probability, maximum distance and scope invalidation [11]. The NNMPCR cache replacement policy (Fig. 5) works as follows:

- All mobile clients generalized movement and access pattern data.
- Mobile clients specific data.
- Predict the next location of client by neural network.
- CMIP associated frequently accessed data items.

```
Client-side cache management algorithm:
Mobile Clients current location and direction is given as
input vector to neural network to predict the future
location.
Mobile client request for data item  $D_i$ 
if  $D_i$  is valid and in cache then
    validate and return the data item  $D_i$ 
end
if cache miss for data item  $D_i$  then
    CMIP for data item  $D_i$  and miss-prefetch set
    to server Get data items from server
end
if enough free space in cache then
    Store data items in cache and update  $pa_i$  of
    accessed data items.
else if not enough free space in cache then
    while ( not enough space in cache)
        if (location proximity & single)
            ANN location proximity replacement
        else
            Replacement based on Invalid scope  $vs_i$ ,
            Minimum  $pa_i$  and Large distance  $ds_i$ 
            If (Multiple replacement )
                Large size with scope invalidation.
            end
            Cache data items and update  $pa_i$  of
            accessed data items.
        end
    end
end
```

Fig. 5. Code for the future direction of mobile client

- If the location in proximity apply a neural network for replacement else normal replacement.

VI. SIMULATION MODEL

NS2 is used to simulate the proposed NNMPCR policy. The network is considered as single, large service area. Seamless hand-off from one cell to another is assumed [3][6]. The mobile clients move freely within the service area to obtain location dependent information. The node density is changed by changing the number of nodes between 5 and 50 within fixed service area. The transmission range of 250m is assumed with wireless bandwidth of 2Mbps [3][6][18]. Initially mobile clients are randomly distributed in the fixed service area. The mobile clients move according to the random waypoint mobility model. Each client chooses a random destination and moves towards with a random velocity chosen from $[V_{min} - V_{max}]$.

After reaching the destination, the client stops for duration defined by the pause time parameter. After this duration, it again chooses a random destination and repeats the whole process again until the simulation ends. The query interval follows an exponential distribution. The mobile client does not issue new query unless the pending query is served. The various parameters of NS2 simulation are listed in Table-I. Normally mobile clients follow longer regular paths and only change in direction is occasional.

TABLE 1. SIMULATION PARAMETERS

Range	Default value	Parameter
Database size	200 data items	Each data item have 4-10 different LDD values
Number of Clients	20	5-50
Client cache size	500 KB	200-2000KB
Client Speed ($V_{min} - V_{max}$)	2 m/s	2-20 m/s
Bandwidth	2 Mbps	
TTL	500 s	200-5000 s
Pause Time	10 s	
Transmission range	250 m	25-250 m
Simulation Time	5000 Sec	2000-10000s
Mobility Model	Random waypoint	
Replacement policy	NNMPCR	LRU, FAR, PPRP

VII. PERFORMANCE EVALUATION

NS2 is used to simulate the proposed NNMPCR policy. The performance of NNMPCR is compared with temporal policy LRU, spatial policy FAR and mobility aware policy PPRP. For implementation of NNMPCR, the database is created with different regions with locations and location specific resources. Data for user movement and [data] access is collected. A data set consists of approximately 1200 records with different data size. The assumptions are:

- Data items are updated only at the server.
- A timestamp and broadcast based cache invalidation method [6] for the mobile clients to maintain cache consistency.

For evaluation, results are obtained when the system has reached the steady state so that the warm-up effect of the client cache is eliminated.

The primary performance metric to evaluate the performance of CRP is a cache hit ratio. Other performance metrics can be derived from it. The ratio shows the number of queries answered locally without sending a request to the server. Thus, if higher the cache hit ratio, higher is the local data availability and less uplink costs. We have conducted experiments by varying the mobile client speed, query interval and cache size. Query interval is the time interval between two consecutive client queries. The cache hit ratio of NNMPCR and other

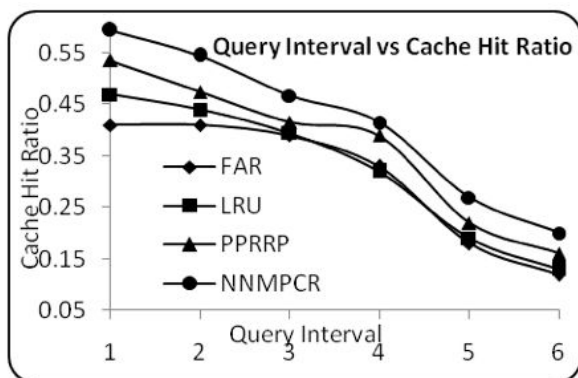


Fig. 6. Cache hit ratio vs Query Interval (sec).

replacement policies with moderate speed is shown in Fig 6. The cache hit ratio decreases with the increase in query interval

since lesser number of queries is executed at each location. Small query interval means more local queries and allows mobile client to fill its cache with more local information quickly, resulting in more cache hits. The LRU policy gives good cache hit ratio with less speed. As speed increase, the temporal locality of access pattern decreases, resulting drop in performance of LRU. The NNMPCR start giving better cache hit ratio by predicting mobile clients future location when making cache replacement decisions. The average performance improvement of NNMPCR is 29.84% as compared to temporal and 14.73% that of mobility aware scheme.

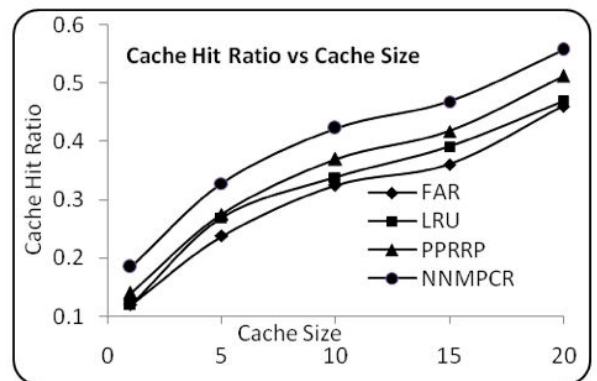


Fig. 7. Cache hit ratio vs Cache size

For small cache size, the average performance improvement of NNMPCR is 16.64% better than the mobility aware scheme and around 8.67% better for large cache size (Fig. 7). With large cache size and query interval the replacement becomes less frequent. This is because the cache can hold a large number of data items which increases the probability of getting cache hit. In such cases what does matter is how accurate our predictions are. So with almost 95% accurate prediction and associative prefetch, hit ratio maintains the consistency. In certain situations, if future location prediction and so prefetching is incorrect then it hampers the performance. Still the overall cache hit ratio is around 8-10% as compared to other mobility aware replacement policies.

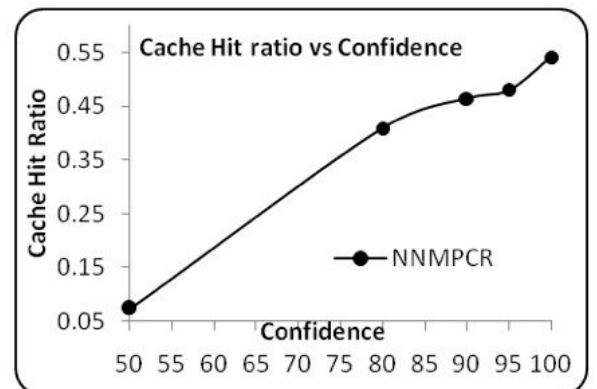


Fig. 8. Cache hit ratio vs Confidence

Fig.8 shows the impact of confidence on cache hit ratio. As expected, cache hit ratio increase with the increase in

confidence. With small confidence, the unassociated data items becomes part of cache-miss set and will be prefetched resulting in very low cache hit ratio.

VIII. CONCLUSION

In this paper, we presented a cache replacement policy using neural network prediction model and cache-miss-initiated prefetch scheme. The neural network predicts the future location of a mobile client based on generalized history of movement patterns. The ANN is designed and trained for single and multiple moves. The NNMPCR considers the temporal and spatial properties of mobile clients access patterns to improve caching performance. CMIP the client-side prefetching scheme is used to prefetch the right data items to reduce future cache misses. The neural network future location prediction, CMIP scheme and location proximity for replacement improves cache hit ratio by manifold. In certain cases if location proximity false then valid scope, data size, data distance and access probability of data item based replacement will take place. Whenever single (data item) storage results into multiple (three or more than three) replacements, we have considered this situation as critical and handled differently. Simulation results for query interval and cache size show that the NNMPCR has significant improvement in performance than the LRU, FAR and PPRRP.

REFERENCES

- [1] D. Barbara, Mobile Computing and Databases: A Survey," IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, pp. 108-117, January/February 1999.
- [2] D. Barbara and T. Imielinski, Sleepers and Workaholics: Caching Strategies in Mobile Environments, In the Proceedings of the ACM SIGMOD Conference on Management of Data, Minneapolis, USA, pp. 1-12, 1994. Pearson: Prentice Hall. p. 6.
- [3] A. Kumar, M. Misra and A.K. Sarje, "A Predicated Region based Cache Replacement Policy for Location Dependent Data In Mobile Environment", I. J. Communications, Network and System Sciences. 2008.
- [4] Ajey Kumar, Manoj Misra and A. K. Sarje, A New Cost Function based Cache Replacement Policy for Location Dependent Data in Mobile Environment, The 5th annual inter research institute student seminar in computer science (IRISS 2006).
- [5] Ajey Kumar, Manoj Misra and A. K. Sarje, A Weighted Cache Replacement Policy for Location Dependent Data in Mobile Environments, SAC07, March 11-15, 2007, Seoul, Korea. Copyright 2007 ACM, pp 920-924.
- [6] K. Lai, Z. Tari and P. Bertok, Location-Aware Cache Replacement for Mobile Environments, IEEE Global Telecommunication Conference (GLOBECOM 04), Vol. 6, pp. 3441-3447, 29th November- 3rd Dec'04.
- [7] Mary Magdalene Jane, R. Nadarajan, Maytham Safar, a spatio-temporal Cache replacement Policy for location Dependent Data in Mobile environments, International Journal of Business Data Communications and Networking, 6(3), 31-48, July-September 2010.
- [8] E. ONeil and P. ONeil, The LRU-k page replacement algorithm for database disk buffering, In the Proceedings of the ACM SIGMOD, Vol. 22, No. 2, pp. 296-306, 1993.
- [9] Q. Ren and M. H. Dhunham, Using Semantic Caching to Manage Location Dependent Data in Mobile Computing, In the Proceedings of 6th ACM/IEEE Mobile Computing and Networking (MobiCom), Boston, USA, pp. 210-221, 2000.
- [10] Kakhkashan Tabassum , Mahmood Quadri Syed and A. Damodaram, Enhanced-Location-Dependent Caching and Replacement Strategies in Mobile Environment, IJCSI Issues, Vol. 8, Issue 4, No 2, July 2011.
- [11] Hariram Chavan , Suneeta Sane , H. B. Kekre, A Markov-Graph Cache Replacement Policy for Mobile Environment, International Conference on Communication, Information and Computing Technology, ICCICT 2012 19-20th October 2012.
- [12] Rooma Rathore, Rohini Prinja, An Overview of Mobile Database Caching, white paper.
- [13] Heloise Mnica, Murilo Silva de Camargo, Alternatives for Ccache Management in Mobile Computing , IADIS International Conference Applied Computing 2004.
- [14] Hui Song, Guohong Cao, Cache-miss-initiated prefetch in mobile environments, Computer Communications www.elsevier.com, 28 (2005) pp. 741-753
- [15] El Garouani Said, El Beqqali Omar, Laurini Robert, Data Prefetching Algorithm in Mobile Environments, European Journal of Scientific Research ISSN 1450-216X Vol. 28 No.3 (2009), pp.478-491
- [16] George Pallis, Athena Vakali, Jaroslav Pokorny, A clustering-based prefetching scheme on a Web cache environment, www.elsevier.com/locate/compeleceng , Computers and Electrical Engineering 34 (2008) 309-323.
- [17] Ajey Kumar, Manoj Misra and A. K. Sarje , Strategies for Cache Invalidation of Location Dependent Data in Mobile Environment, Department of Electronics and Computer Engineering IIT Roorkee, Roorkee, India.
- [18] B. Zheng, J. Xu and D. L. Lee, Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments, IEEE Transactions on Computers, Vol. 51, No. 10, pp. 1141-1153, Oct'02.
- [19] Alok Madhukar, Tansel O zyer, Reda Alhadj, Dynamic cache invalidation scheme for wireless mobile Environments, Springer Science+Business Media, LLC 2007, pp. 727-740.
- [20] Anil K. Jain, Jianchang Mao, K.M. Mohiuddin, Artificial Neural Networks: A Tutorial, 1996 IEEE, pp. 31-44.
- [21] Jens Biesterfeld, Elyes Ennigrou, Klaus Jobmann, Neural Networks for Location Prediction in Mobile Networks, Institut fr Allgemeine Nachrichtentechnik, Universitt Hannover D-30167 Hannover, Germany.
- [22] B. P. Vijay Kumar, P. Venkataram, Prediction-based location management using multilayer neural networks", J. Indian Inst. Sci., 2002, 82, 7-21, Indian Institute of Science.
- [23] Joe Capka and Raouf Boutaba, Mobility Prediction in Wireless Networks Using Neural Networks, J. Vicente and D. Hutchison (Eds.): MMNS 2004, LNCS 3271, pp. 320-333, 2004.
- [24] Pamler, Hertz, Krogh, Introduction to the theory of Neural Network, pp.3-15,115-161.
- [25] Jean-Marc Francois, Performing and Making Use of Mobility Prediction PhD Thesis 2006-07.
- [26] Tong Chang, Analysis of critical success factors of mobile location-based services, Thesis, June 20, 2009.
- [27] Holger Kirchner, Reto Krummenacher, David Edwards-May and Thomas Risse, A Location-aware Prefetching Mechanism, Fraunhofer IPST, Darmstadt Germany, Euromapping, Seyssinet, France.
- [28] Lucian Vintan, Arpad Gellert, Jan Petzold, and Theo Ungerer"Person Movement Prediction Using Neural Networks ",In First Workshop on Modeling and Retrieval of Context, Ulm, Germany, September 2004.
- [29] Satish Kumar, Neural Networks - A Classroom Approach, TMH, pp. 164-175.
- [30] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, Mining Association Rules between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD Conference Washington DC, USA, May 1993.