The CSI Journal on
Computer Science and Engineering
Vol. 19, No. 2, 2025
Pages 19-26
doi.org/10.22034/jcse.2025.181978
Regular Paper

Smart ATM Cash Replenishment Planning using Deep Reinforcement Learning

Mohammadhossein Kiyaei [1] Farkhondeh Kiaee [2]

[1] Finance Dept., Faculty of Management & Accounting Farabi Campus, University of Tehran Qom, Iran
[2] Dept. of Electrical & Computer Engineering, Faculty of Shariaty Technical and Vocational University (TVU) Tehran, Iran

Abstract

Access to cash for many in society is remaining essential during the current COVID-19 lock-down around the globe. A smart city requires the banking industry to exploit IoT and Artificial Intelligence (AI) in order to track its ATM network and predict outages due to cash shortages. In this paper, we study the real-time cash replenishment planning problem under outflow uncertainty where the fee of the security companies grows if the replenishment ends up falling on a weekends/holiday. Our model is based by the Double Deep Q-Network (DQN) algorithm which combines popular Q-learning with a deep neural network. The proposed method is used to minimize the ATM replenishment cost where the cash demand changes dynamically at each day. The performance analysis of the proposed method for different amounts of replenishment cash shows that the the proposed method can effectively work under real word conditions and reduce the ATM operational cost compared with the other state- of-the-art cash demand prediction schemes.
Keywords: cash replenishment planning, deep learning, ATM, reinforcement learning, double Q-network.

1. Introduction

ATMs are no longer just machines, these connected devices are smart, intelligent things in the Internet of Things (IoT). They are used by the majority of the costumers to withdraw cash. They are playing an even more critical role in ensur- ing that consumers have access to cash and wider banking services while branches have reduced hours or closures, or customers want to avoid face-to-face or in branch interactions completely. The modern banking industry that emerges from the COVID-19 crisis embraces the Internet of Things (IoT) and Artificial Intelligence (AI) technologies that will entirely replace humans by taking management decisions.
Cash replenishment planning (CRP) system helps formulate a cost-efficient operating plan that includes the fee for secure ATM replenishment service. The overview of the system is shown in Fig. 1. The ATMs using the IoT technology are linked with the monitoring center, which enables automatically analysing the cash remaining in the ATM and interacting with the security company necessary for replenishment plan execution.

CRP problem for ATMs has been extensively investigated by researchers in the economic and financial management community. We may classify the developed approaches ac- cording to their underlying assumptions into two main groups. The first group assume the future ATM cash withdrawal

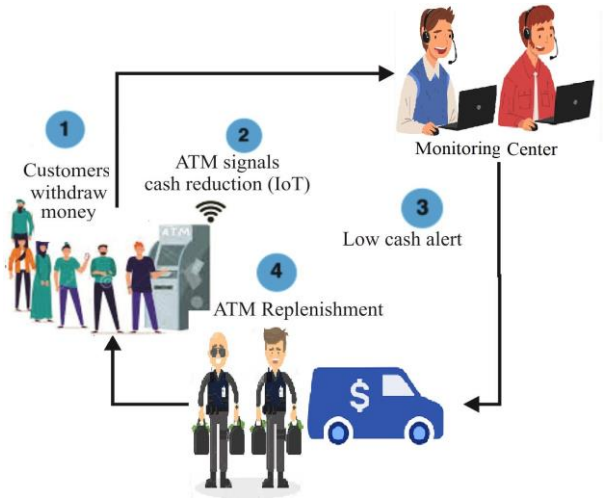

Fig. 1: Overview of the CRP system.

information, is known in advance and thus are not real- time. These methods seek for an optimization method towards optimum planning of replenishment, namely particle swarm optimization [1], Heuristic Optimization [2], and decision trees [3]. Several optimal CRP solutions based on dynamic programming (DP) algorithm are proposed in literature [4], [6] The second group is more challenging due to the lack of cash demand information. The models in this group are mainly based on the learning systems and use the historical data to predict the future required information (prediction-based CRP). In particular, an AI model namely, feed-forward neural networks (NNs) [7], [8], recursive NNs [9], [10], or combination of clustering and NNs [11], [12] is trained to make real-time decisions.

In this paper, an AI decision-making model based on reinforcement learning (RL) is introduced. In the proposed model, an agent explores the unknown CRP environment and learns the value associated to each action taken in a different set of states. The instructed action-value function defines a decision criterion which helps to take the optimum actions in an immediate (real-time) manner [13].

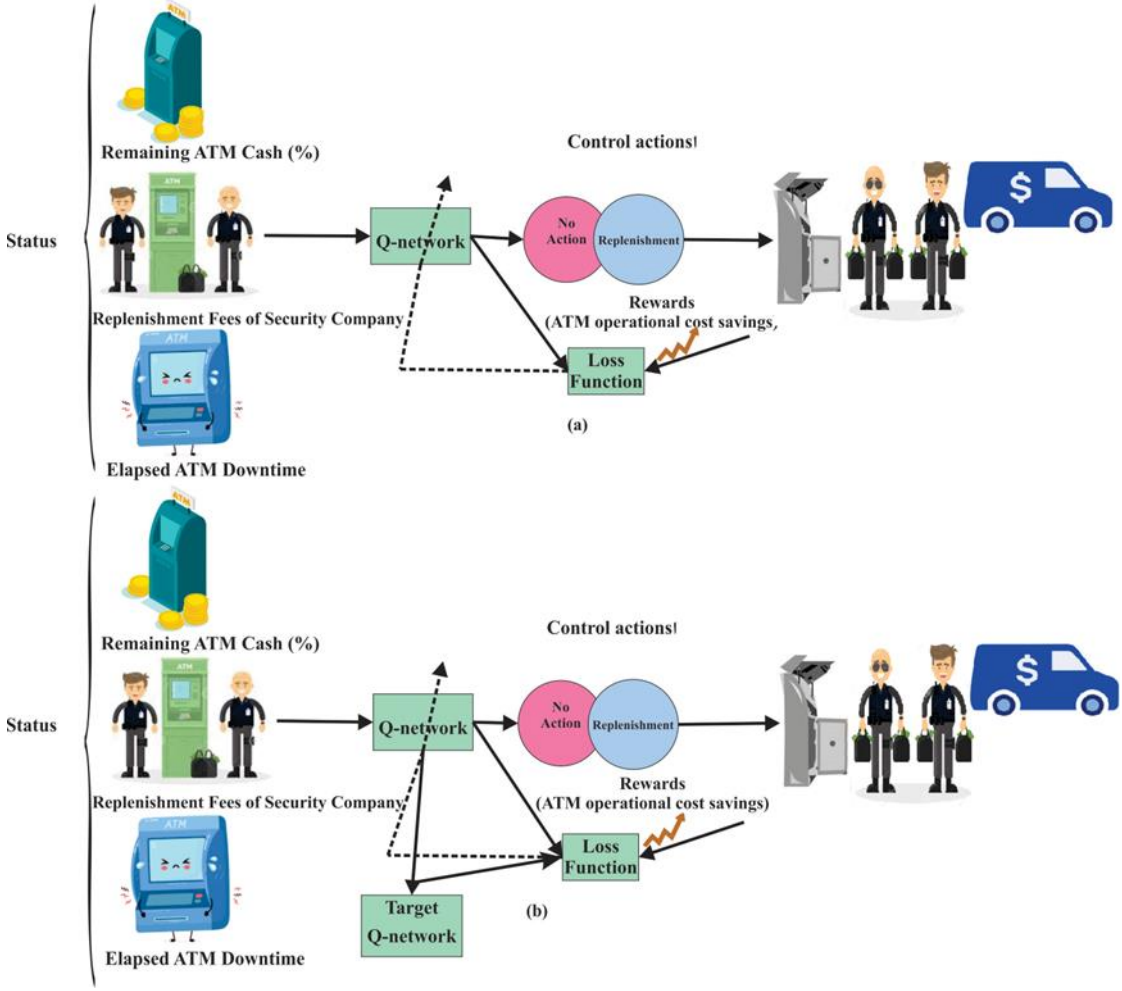

Fig. 2: Schematic of the RL-based CRP system. (a) DQN , (b) Double DQN.

In particular, in this paper, the deep Q-learning as the most established realization of RL approaches, is applied to the CRP system. Our goal is to build a smart system that determines when to perform replenishment, based on the current and historical system information.

For systems with continuous observation, the application of the neural networks to Q-learning, termed a Q-Network [13] is developed. Deep learning is the term given to neural networks with many layers, and has been shown to be effective in learning high level features from large input spaces [14], [15]. The training instability of Q-learning is addressed by the introduction of Double variant of Deep Q-Networks [16]. The Double DQN method effectively reduces the correlations of the action-values with the target [17].

In this paper a Double DQN+IoT method is proposed which benefits from both IoT and AI technologies. The method is tested on the real *NN5* dataset that contains daily ATM cash withdrawal amounts over 2 years across the US. the proposed method can efficiently model cash demand uncertainty and take into account replenishment cost variability in order to reduce ATM operational cost.

The proposed Double DQN+IoT system is compared with the system based solely on the IoT, shallow Q-network, and prediction based NN methods under diverse testing conditions. In the experiments, the impact of different amounts of the total replenishment cash on the performance of the methods is in- vestigated. The total amount of cash that can be used for ATM operation is arranged with the financial institution in advance. If cash is replenished to the ATM's full capacity, number of replenishments can be significantly reduced. However, it will increase the unused cash in the ATM resulting in waste of funds. The experimental results show that the proposed method can make reliable cost savings under real ATM outflow dataset. The outline of this paper is as follows. Section II presents a general formulation of the CRP problem. Section III describes the deep Q-learning approach and presents the implementation details of the Double DQN CRP system. The experimental results and performance comparison with other methods are
presented in Section IV. Finally, V concludes the paper.

Problem Formulation

The aim of CRP system is to reduce replenishment cost through efficient planning that benefits both the financial institutions and customers by reducing the ATM downtime without increasing ATM operational costs.

Let $0 \leq b_t \leq 1$ be the percentage of the remaining ATM cash that is available at time point t and l_t denote the elapsed downtime after ATM runs out of money. If the ATM runs out of cash, customers are dissatisfied due to bad service. The maximum allowable ATM downtime due to lack of cash is then restricted to be D_{max} days. It is assumed that the contract of financial institution with the security company is on a per replenishment basis without a fixed fee for all week days. The cost of replenishment runs on business days is 10$. However, if replenishment day falls on a weekend/holiday, the security company performs replenishment by the fee of 30$.

The state space of the cash replenishment planning problem comprises of the ATM remaining cash space, the replenishment cost space and the elapsed ATM downtime space. The state of the system at time t is then defined as $s_t = [b_t, p_t, l_t]$ The action in the cash replenishment planning can be interpreted as choosing one operation from the action space A = {replenishment, no-action}. However, due to the constraints in the problem, not all the actions can be performed at a given state. The set of all possible actions given the state of the system is limited by the maximum allowable ATM downtime due to lack of cash D_{max}. Taking "no action" is then not allowed when $l_t > D_{max}$.

Reward function is a key ingredient of the reinforcement learning systems. The reward is defined as the ATM operational cost savings for the financial institutes. The corresponding rewards for the replenishment action is then negative of the money paid by the financial institute to the security company. However, the corresponding rewards for the no (replenishment) action is positive, as the financial institute saves money by reducing ATM operational cost.

Algorithm 1 Training process of the Double DQN CRP system
function Double DQN CRP $\{s_i = [p_i, b_i, l_i],\ i = 1, \ldots, t_d\}$

Initialize replay buffer B to a defined capacity.
Initialize action-value Q-function with random weights θ
Initialize target action-value $\tilde{Q}$-function with weights $\tilde{\theta} \leftarrow \theta$
for episode 1,M **do**
    Receive initial remaining ATM cash sequence b_1 and form initial state $s_1 = [b_1, p_1, l_1]$
    for t steps **do**
        With probability ϵ select a random action, otherwise select $a_t = \arg max_a Q(s_t, a; \theta)$
        Execute a_t and observe reward r and new ATM cash sequence p_{t+1} (next state: $s_{t+1} = [b_{t+1}, p_{t+1}, l_{t+1}]$)
        Store transition $\{s_t, a_t, r_t, s_{t+1}\}$ in buffer
        Sample random batch of transitions from B
        Perform a gradient descent step on (4) with respect to the network parameters θ
        Update the target network using (5).
    end for
end for
Return θ, $\tilde{\theta}$

3. Double DQN Cash Replenishment Planning System

Reinforcement learning (RL) is a general framework to deal with sequential decision tasks. Fig. 2 shows the schematic of the CRP system using reinforcement learning with DQN method and its Double DQN extension.

At each time step t, RL observes the status s_t of the environment, takes an action a_t, and receives some reward r_t from the environment. The RL method suggests that, given sufficient pairs of (s_t, a_t, r_t), the optimal policy Q^* is to maximize the long-term accumulated reward

$$Q^*(s, a) = \max_{\pi} E_{\pi}\{R_t \mid s_t = s, a_t = a\}. \quad (1)$$

The Q-function holds Bellman equation property formulated as:

$$Q^*(s_t, a_t) = r + \gamma \max_a Q^*(s_{t+1}, a) \quad (2)$$

For systems with continuous state s, a neural network is often used to approximate the value $Q(s, a)$. This network is often referred as a Q-network [13]. If the Q-network consists of several layers of nodes, we obtain the deep Q-learning architecture. Deep learning has been known to have the ability to learn hierarchical patterns, and the patterns learned by the upper layers tend to be abstract and invariant against disturbance. The Q-network is trained to minimize the Q prediction error, i.e., the difference between the left-hand and right-hand side of Eq. 2. The loss function is then formulated as follows:

$$\min_{\theta} L(\theta) = \sum_{i \in V} \left(y_i - Q(s_i, a_i \mid \theta)\right)^2,$$
$$y_i = r_i + \gamma \max_a \tilde{Q}(s_{i+1}, a \mid \theta), \quad (3)$$

where i and θ denote the training iteration and the parameters of the Q-network, respectively. The training examples are in the form of (s_i, a_i, r_i, s_{i+1}), and B denotes the buffer

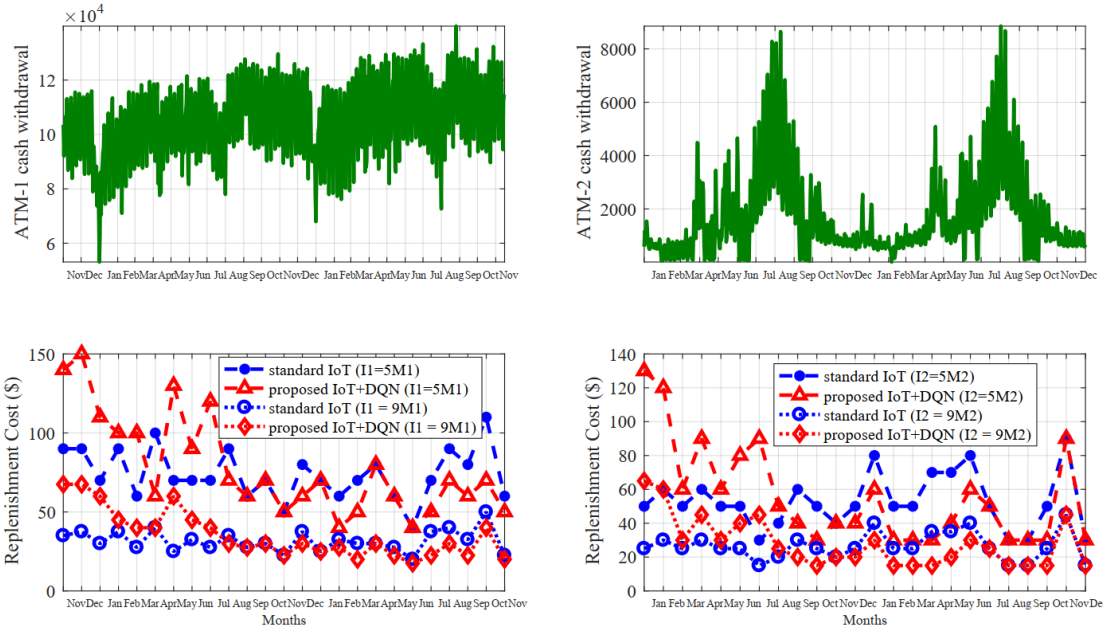

Fig. 3: Top: Daily cash flow of the two ATMs (NN5 dataset). Bottom: Performance of the proposed Double DQN+IoT system vs the system based solely on the IoT for different replenishment amounts.

containing the recent training examples. In addition, y_i is the prediction of $Q(s, a)$ given by the Bellman Eq. 2.

The deep loss functions are typically minimized using stochastic gradient descend (SGD) algorithm. The gradient of Eq. 3 with respect to θ is given by

$$\nabla_\theta L = \sum_{i \in V} \left(y_i - Q(s_i, a_i \mid \theta)\right) \nabla_\theta Q(s_i, a_i \mid \theta) \quad (4)$$

Where $\nabla_\theta Q(s_i, a_i \mid \theta)$ can be easily computed by the back-propagate (BP) algorithm.

We avoid the divergence of direct implementation of the system with neural networks due to using the same Q-network in calculating the target value y_i in (3). Our solution is similar to the target network used in Fig. 2 for Q-learning. The authors in [16] show that stable targets y_i are required in order to train the system, consistently. A copy of the Q-network (represented by $\tilde{Q}$) is then created and used for calculating the target values. The weights of the target $\tilde{Q}$ network (indicated by $\tilde{\theta}$) are softly updated by interpolating with the latest θ, as follows:

$$\tilde{\theta} = \tau\theta + (1 - \tau)\tilde{\theta}, \quad (5)$$

where τ is the interpolation factor. The relatively unstable problem of learning the action-value function then approaches to a case of robust supervised learning problem. Although, the delays in updating the target values may slow learning, in practice the stability of learning is greatly outweighed. Note that the decision of the replenishment is made based on the target network $\tilde{Q}$, rather than the present network Q. An overview of the Double-DQN method for CRP system is outlined in Algorithm. 1.

4. Performance Evaluation

In this section, the performance of the proposed RL-based CRP system is evaluated using the actual ATM cash with-drawal *NN5* dataset [12]. Two ATMs with different daily outflow characteristics are selected from *NN5* dataset. The cash demand time-series of the selected ATMs are illustrated in Fig.

3. The ATM-1 dataset consists of historical daily outflow from 01 Nov 2003, to 16 Nov 2005. The dataset corresponding to ATM-2 includes daily outflow form 01 Jan 2002 to 17 Dec 2003. The average daily cash demand of the ATM-1 and ATM- 2 is M_1 = \$106,000 and M_2 = \$1,650, respectively.

Number of times an ATM is replenished ties closely with the amount of cash that is replenished. We assume that a fixed amount of cash is loaded into the ATM at each replenishment action. Let I_1 and I_2 denote the replenishment amount for ATM-1 and ATM-2, respectively. The replenishment amount should be established so that it remains within the amount predicted for a certain period. However, due to ineffective cash forecasting, the financial institutions often maintain more cash than necessary at their ATMs. The average monthly replenish- ment cost of the proposed method for different replenishment amounts, $I_1 \in \{5M_1, 9M_1\}$ and $I_2 \in \{5M_2, 9M_2\}$ is provided in Fig. 3.

The maximum allowable ATM downtime due to lack of cash, D_{max}, is selected to be 2 days. Provided in Fig. 3 is also the results of the standard CRP method based solely on the IoT, in which the ATM remaining cash (%) is signalled to the monitoring center by the smart ATM. Therefore, replenishment is done if the elapsed ATM downtime is more than D_{max} days. If the replenishment ends up falling on a weekend i.e.

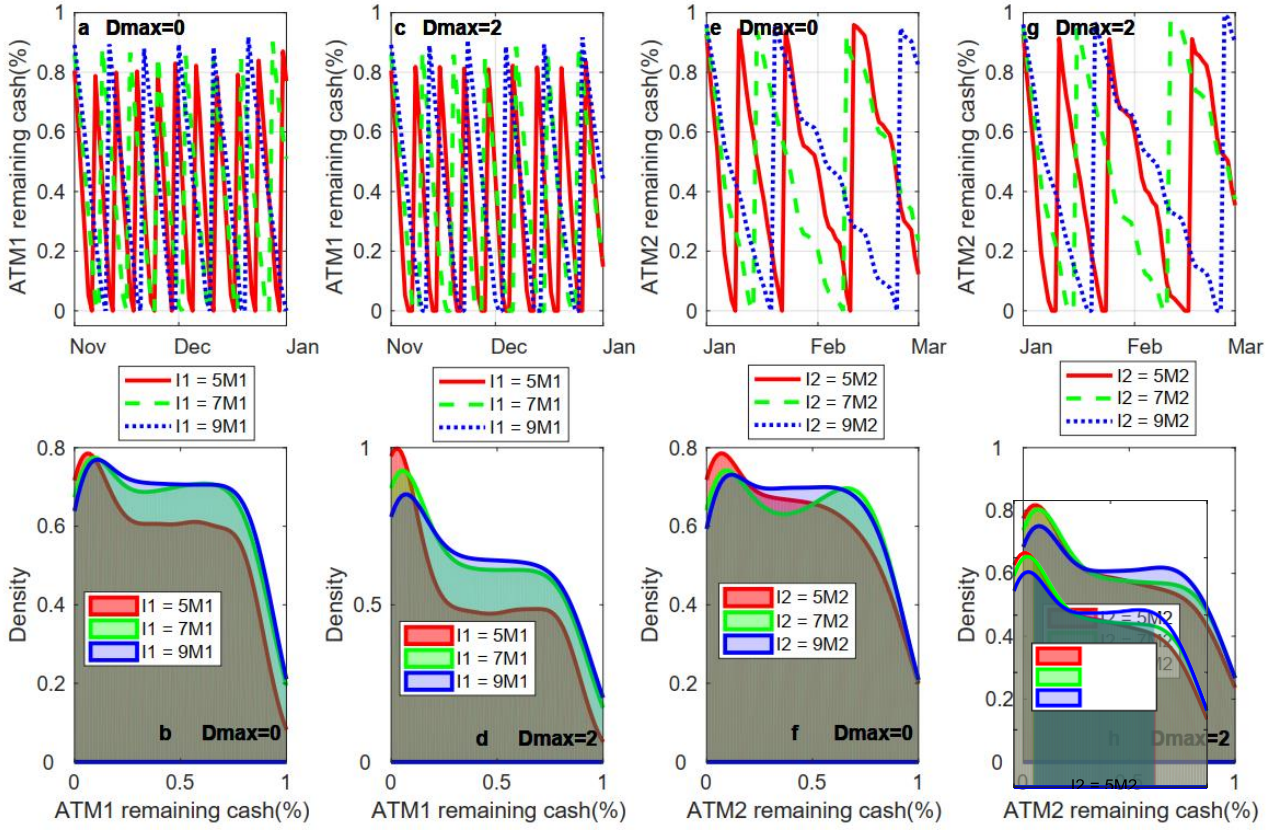

Fig. 4: Percentage of the ATMs' remaining cash (Top) and its density distribution (Bottom) using the proposed IoT+DQN method under different replenishments amounts I and maximum allowable downtimes D_{max}.

Saturday and Sunday, the fee of the security companies is considered to be three times greater than business days (e.g., 30$ for weekends vs 10$ for business days is considered here for the replenishment cost computations).

From Fig. 3, it can be seen that as the proposed DQN algorithm learns from experience and adapts to different cash demands and variable costs, the monthly cost of the proposed algorithm begins to decrease. From the results, it is observed that after the initial training, which takes about 12 months, the proposed Double DQN+IoT system makes more cost savings than the system based solely on the IoT.

The experiment is repeated using the proposed Double DQN+IoT method under different replenishments amounts $I \in \{5M, 7M, 9M\}$ and maximum allowable downtimes $D_{max} \in \{0, 2\}$. The ATM-1 (ATM-2) cash levels (%) during the first two-month period for different D_{max} values is pre- sented in Fig. 4.a and 4.c (Fig. 4.e and 4.g). It can be observed that the ATMs' emptying rate between two adjacent replenish- ments (peak points) follows a line or a simple polynomial pattern. The results also show that for $I = 9M$, the number of replenishments the security companies need to perform is reduced when compared with $I = 7M$ or $I = 5M$ at the same period.

Density distribution of the remaining cash ratios for the ATM-1 (ATM-2) is visually presented in Fig. 4.b and 4.d (Fig. 4.f and 4.h). Results show that $I = 5M$ has higher density in the lower cash levels than $I = 9M$ because reducing replen- ishment amount I increases the risk of running out of cash. As the replenishment amount, I increase, the density profile tends to become flat. The extreme completely flat density happens only when one replenishment is performed during the total period. When comparing the emptying behaviour across the selected two D_{max} values, the density of $D_{max} = 2$ shows a more lower value concentration compared to $D_{max} = 0$. This is due to the fact that the recurrence of lower cash levels is increased by increasing maximum ATM downtime due to lack of cash.

Fig. 5.a shows, the actual actions taken in the standard IoT method and the proposed Double DQN+IoT method on different days of the week corresponding to ATM-1 data during June and July 2005. As indicated by the arrow in certain places in the figure, in the second scenario the replenishment operation is not postponed to the 6th and 7th

days of the week, which successfully reduces the ATM operational cost.

The total replenishment actions are then decomposed based on their week days in Fig. 5.b in order to illustrate the weekly distribution pattern in the results. It can be seen from Fig. 5.b that for the proposed Double DQN+IoT system the concentration of replenishment actions predominately is decreased in Saturday and Sunday and is shifted around business days. For the standard system based solely on the IoT, it is observed that the weekly pattern vanishes and the behaviour is more the same across different days. The explanation of this effect is that the proposed method starts learning more from both

TABLE I: Performance comparison of the proposed Double DQN+IoT and other existing methods. The average monthly cost along with the standard deviation are summarized (One-tailed p-values of two sample t-tests for a null hypothesis that the value of performance of Double DQN+IoT is larger than that of the other methods are presented in parenthesis)

			IoT+DQN	(Based solely on) IoT	IoT+SQN	IoT+CDNN	IoT+LSTM	IoT+RNN	IoT+SNN
ATM-1	$Dmax=0$	$I1 = 5M_1$	120.1 ± 3.6	147.3 ± 4.7 (0.000)	132.1 ± 5.1 (0.008)	136.1 ± 6.02 (0.000)	130.6 ± 4.3 (0.012)	143.2 ± 9.8 (0.000)	146.8 ± 10.9 (0.000)
		$I1 = 7M_1$	83.8 ± 2.1	103.1 ± 3.1 (0.000)	92 ± 3.2 (0.009)	94.3 ± 3.9 (0.000)	90.8 ± 3.8 (0.023)	99.7 ± 6.5 (0.000)	102 ± 9.1 (0.000)
		$I1 = 9M_1$	49.5 ± 1.3	61.7 ± 1.9 (0.000)	54.6 ± 2.2 (0.010)	56.5 ± 2.6 (0.000)	54.0 ± 2.4 (0.014)	59.5 ± 4.0 (0.000)	61.0 ± 4.3 (0.000)
	$Dmax=2$	$I1 = 5M_1$	86.0 ± 3.2	105.9 ± 3.4 (0.000)	94.4 ± 3.6 (0.006)	97.5 ± 4.3 (0.000)	93.3 ± 3.1 (0.010)	102.3 ± 7.3 (0.000)	104.9 ± 8.1 (0.000)
		$I1 = 7M_1$	67.1 ± 1.6	82.5 ± 2.5 (0.000)	73.6 ± 2.9 (0.008)	75.5 ± 3.1 (0.000)	72.7 ± 3.0 (0.021)	79.8 ± 5.2 (0.000)	81.6 ± 7.3 (0.000)
		$I1 = 9M_1$	45.0 ± 1.1	56.1 ± 1.8 (0.000)	49.7 ± 2.1 (0.009)	51.4 ± 2.5 (0.000)	49.1 ± 2.1 (0.011)	54.1 ± 3.8 (0.000)	55.5 ± 4.2 (0.000)
ATM-2	$D_{max}=0$	$I2 = 5M_2$	161.5 ± 2.1	184.8 ± 5.9 (0.000)	167.4 ± 3.1 (0.012)	175.9 ± 5.1 (0.000)	169.7 ± 3.3 (0.019)	179.1 ± 6.6 (0.000)	181.3 ± 7.1 (0.000)
		$I2 = 7M_2$	112.6 ± 1.5	128.5 ± 2.9 (0.000)	117.7 ± 2.1 (0.009)	107.7 ± 3.9 (0.000)	118 ± 2.7 (0.015)	124.5 ± 4.3 (0.000)	126.1 ± 3.7 (0.000)
		$I2 = 9M_2$	67.1 ± 1.2	77.6 ± 2.4 (0.000)	71.2 ± 1.9 (0.011)	73.8 ± 2.4 (0.000)	72.2 ± 2.1 (0.021)	75.2 ± 3.2 (0.000)	76.1 ± 2.9 (0.000)
	$D_{max}=2$	$I2 = 5M_2$	115.4 ± 1.4	132 ± 3.9 (0.000)	119.66 ± 2.5 (0.008)	125.7 ± 3.9 (0.000)	121.2 ± 2.6 (0.017)	127.9 ± 5.6 (0.000)	129.5 ± 5.1 (0.000)
		$I2 = 7M_2$	90.1 ± 1.3	102.8 ± 2.7 (0.000)	94.2 ± 1.9 (0.006)	97.9 ± 3.1 (0.000)	94.4 ± 2.6 (0.013)	99.6 ± 4.1 (0.000)	100.8 ± 3.5 (0.000)
		$I2 = 9M_2$	61.9 ± 1.1	70.6 ± 2.2 (0.000)	64.7 ± 1.8 (0.007)	67.1 ± 2.2 (0.000)	65.6 ± 1.9 (0.019)	68.3 ± 3.1 (0.000)	69.2 ± 2.8 (0.000)

cash demand data and replenishment fees which are higher on two last non-working days of the week. In other words, the proposed Double DQN+IoT is shifted replenishment actions to be less for two weekend days due to their higher fees in the analysed scenario.

We compare the proposed IoT+DQN system with the performance of Shallow Q-network (SQN). Moreover, the performance of proposed method is compared with the prediction-based NNs namely, Shallow Neural Network (SNN), Convolutional Deep Neural Network (CDNN), RNN and long short-term memory (LSTM).

The goal of the prediction-based NN is to predict whether

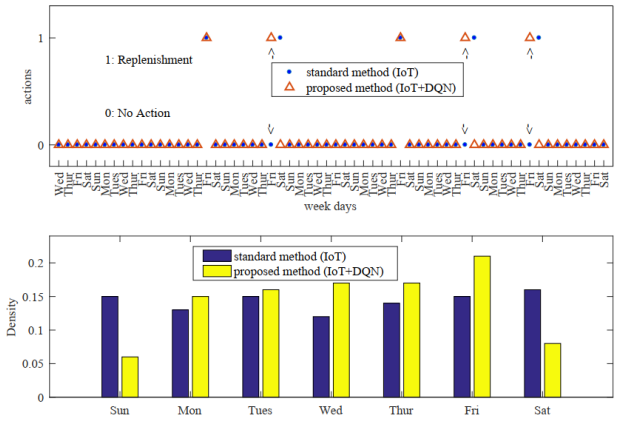

Fig. 5: Review of the actions taken by the systems (a) details during June & July (b) weekly distribution of total actions.

the cash demand of the next day is going to surpass the ATM remaining cash which corresponds to taking replenishment action. Two models based on shallow networks i.e., SQN and SNN is considered where a network with two layers is combined with clustering and feature selection [18]. A small

representation of data is first achieved using clustering. A sequential stepwise process, called Backward Greedy Selection, is then used to remove variables (features) that are irrelevant to the neural network performance. A sigmoid transfer function is used in the hidden layer of the network and the output layer is linear, trained with the Levenberg-Marquardt algorithm.

The python-based DL package tensorflow is used to implement the deep CRP system structures. Tensorflow provides the benchmark implementations of convolution, pooling and fully-connected layers for public usages. The proposed DQN is composed of five layers: 1) an input layer (28-dimension input composed of the 26 ATM remaining cash of 26 days in the past along with their corresponding replenishment cost and the elapsed ATM downtime); 2) a convolutional layer with 32 convolutional kernels (the length of each kernel is considered to be 6); 3) a max pooling layer; 4) a fully connected layer; and 5) a soft-max layer with two outputs. The RNN contains an input layer, a dense layer (32 hidden neurons), a recurrent layer, and a soft-max layer for classification. The LSTM shares similar configuration to RNN except for replacing the recurrent layer with the LSTM module.

The training strategy of the deep networks involves iterative updating of the weights in an online manner. In practice, the first 200 time points are used to set up the network weights. At each day, a new training example (s_t, a_t, r_t, s_{t+1}) is added to a defined buffer B (with a finite capacity) consisting of recent CRP system history. The examples in the buffer are used as a mini-batch to train the Q-network following Eq. 3. The trained system is then used to control the CRP system from 201 to 250. In the next iteration, the sliding window of the training data is moved 50 ticks forward covering a new training set from 50 to 250. The parameters in the network are then iteratively updated with the recently released data. This online strategy allows the model to get aware of the latest CRP system condition and update its parameters accordingly.

The average monthly replenishment costs of the proposed algorithm for two ATMs are shown in Table I (Note that the time points of the input time series before convergence employed for system initialization and is not used for performance calculation).

The results of the methods when the replenishment amount, I vary from $5M$ to $9M$ and the maximum downtime D_{max} is set to 0 and 2 are shown in Table I. The experiment is repeated for the 10 different seeds of generating initial random weights and the mean performance is reported. As presented in Table I, all of the performance measures are consistently better for the proposed Double DQN+IoT compared with other methods and the t-test p-values (in parentheses) demonstrated that the differences are all statistically significant.

It can be seen from Table I, the average monthly cost decreases as the replenishment amount increases while D_{max} stays the same. This is due to the fact that less replenishment actions are required in order to meet the increasing cash amount, which results in more cost savings. If we compare the result of different D_{max} at the same replenishment amount, it can be found that increasing the maximum allowable downtime D_{max} can result in an decreasing monthly cost. It is not hard to understand this result as the required replenishments diminish when the allowable downtime is increased.

The results in Table I shows that for both ATMs, the lowest replenishment costs are made by Double DQN+IoT system. This is due to its novel structure which allows simultaneous environment sensing and optimum action learning for CRP system.

When considering the results of CDNN, RNN, LSTM and SNN, the pitfalls of prediction-based NN methods become immediately apparent. By investigating the total profit values in Table I, only the LSTM achieves comparable cost with the other RL-based systems. This is because prediction-based systems only consider the cash demand data to make decisions. The Double DQN learns both cash demand condition and the action-value function $Q(s; a)$ in a joint framework.

5. Conclusion

Modern IoT and AI technologies allow a bank to track its ATM network to help predict outages due to cash shortages. In this work we presented a CRP system based on the deep Q-network (DQN) structure. The system is composed of two main components: a deep learning component that learns the cash demand dynamic status, and a Q-learning component that learns the action-value function. However, the two components are integrated as one, in the real implementation of the system. In order to obtain stable targets during temporal difference calculations, a separate target network is attached to the system thereby forming the final Double DQN structure. The performance of the proposed method under diverse testing conditions i.e. for various amounts of total replenishment cash and different maximum allowable downtime (due to lack of cash) is analysed. Experimental results show that the proposed method outperforms the other state-of-the-art deep CRP systems. The results on real cash demand time-series demonstrate the effectiveness of the learning system in joint system dynamic acquisition and optimal action learning.

References

[1] Y. Li, H. Sun, C. Zhang, and G. Li, "Sites selection of ATMs based on particle swarm optimization," in *2009 International Conference on Information Technology and Computer Science*, vol. 2, IEEE, 2009, pp. 526–530.

[2] V. Platonova, E. Gubar, and S. Kukkonen, "Heuristic optimization for multi-depot vehicle routing problem in ATM network model," in *Advances in Dynamic Games*, Springer, 2020, pp. 201–228.

[3] M. Zeydan and S. Kayserili, "A rule-based decision support approach for site selection of automated teller machines (ATMs)," *Intelligent Decision Technologies*, vol. 13, no. 2, pp. 161–175, 2019.

[4] F. Ozer, I. H. Toroslu, P. Karagoz, and F. Yucel, "Dynamic programming solution to ATM cash replenishment optimization problem," in *International Conference on Intelligent Computing & Optimization*, Springer, 2018, pp. 428–437.

[5] S. Bati and D. Gozupek, "Joint optimization of cash management and routing for new-generation automated teller machine networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[6] C. Bilir and A. Doseyen, "Optimization of ATM and branch cash operations using an integrated cash requirement forecasting and cash optimization model," *Business & Management Studies: An International Journal*, vol. 6, no. 1, pp. 237–255, 2018.

[7] S. P. Arabani and H. E. Komleh, “The improvement of forecasting ATMs cash demand of Iran banking network using convolutional neural network,” *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3733–3743, 2019.
[8] S. Vangala and R. Vadlamani, “ATM cash demand forecasting in an Indian bank with chaos and deep learning,” *arXiv preprint arXiv:2008.10365*, 2020.
[9] S. P. Arabani and H. E. Komleh, “The optimization of forecasting ATMs cash demand of Iran banking network using LSTM deep recursive neural network,” *Journal of Operational Research in Its Applications (Applied Mathematics) - Lahijan Azad University*, vol. 16, no. 3, pp. 69–88, 2019.
[10] H. Abbasimehr, M. Shabani, and M. Yousefi, “An optimized model using LSTM network for demand forecasting,” *Computers & Industrial Engineering*, p. 106435, 2020.
[11] P. K. Jadwal, S. Jain, U. Gupta, and P. Khanna, “K-means clustering with neural networks for ATM cash repository prediction,” in *International Conference on Information and Communication Technology for Intelligent Systems*, Springer, 2017, pp. 588–596.
[12] K. Venkatesh, V. Ravi, A. Prinzie, and D. Van den Poel, “Cash demand forecasting in ATMs by clustering and neural networks,” *European Journal of Operational Research*, vol. 232, no. 2, pp. 383–392, 2014.
[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[14] F. Kiaee, C. Gagné, and M. Abbasi, “Alternating direction method of multipliers for sparse convolutional neural networks,” *arXiv preprint arXiv:1611.01590*, 2016.
[15] F. Kiaee, H. Fahimi, and H. Rabbani, “Intra-retinal layer segmentation of optical coherence tomography using 3D fully convolutional networks,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 2795–2799.
[16] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” *arXiv preprint arXiv:1509.06461*, 2015.
[17] F. Kiaee, “Integration of electric vehicles in smart grid using deep reinforcement learning,” in *2020 11th International Conference on Information and Knowledge Technology (IKT)*, IEEE, 2020, pp. 40–44.
[18] K. L. López, C. Gagné, and M.-A. Gardner, “Demand-side management using deep learning for smart charging of electric vehicles,” *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2683–2691, 2018.