

PDRL-PQCS: Proactive Deep Reinforcement Learning for Real-Time PQC Configuration in ITS with Edge Collaboration

Tongmao Ma, Shanchen Pang, Lu Liu, Xiao Chen, Cong Liu, and Tong Ding*

Abstract The emergence of quantum computing presents significant challenges for intelligent transportation systems (ITS), which rely on traditional public-key cryptography, demanding both strong security and low-latency operation. This highlights an urgent requirement for post-quantum cryptographic (PQC) algorithms to guarantee long-term protection. However, a direct integration of PQC into ITS may introduce latency and inefficiency, as static implementations cannot adapt to dynamic traffic conditions and may apply excessive security in low-risk contexts, resulting in unnecessary computational overhead, particularly within resource-constrained edge computing environments. To mitigate these issues, the PDRL-PQCS (Predictive Deep Reinforcement Learning for Post-Quantum Cryptography Selection) framework is proposed. This approach employs a Transformer-based module to generate multi-step predictions of traffic flow and computational load, thereby providing proactive contextual awareness. These predictions subsequently inform a deep reinforcement learning (DRL) agent, which dynamically selects optimal parameter configurations for the CRYSTALS-Kyber PQC algorithm in real time, enabling adaptive security-performance balancing across the heterogeneous nodes of an ITS edge computing network. Through this two-stage approach, PDRL-PQCS adapts PQC parameters within defined security bounds to optimally balance security assurance with operational efficiency. An extensive experimental evaluation assesses the framework under three representative traffic scenarios with diverse resource availability and congestion levels. Results indicate that PDRL-PQCS achieves 10%–20% higher overall performance compared to baseline methods across various challenging conditions.

Keywords post-quantum cryptography, intelligent transportation systems, reinforcement learning, time-series forecasting, edge computing.

1 Introduction

The swift advancement of technologies such as Internet of Things (IoT), artificial intelligence (AI), and edge computing is transforming Intelligent Transportation Systems (ITS) from isolated vehicular communications into fully integrated networks that link vehicles, infrastructure, and cloud platforms [1, 2]. Within such cooperative settings, safety-critical applications, including cooperative collision avoidance, traffic-signal optimisation, and autonomous driving, necessitate the real-time exchange of vehicle-sensed data and roadside information [3]. These applications require not only dependable communication but also strong end-to-end cryptographic safeguards [4]. Nevertheless, guaranteeing the confidentiality, reliability, and validity of data shared across heterogeneous and open vehicular networks poses significant security challenges, especially given the imminent threat presented by quantum computing [5].

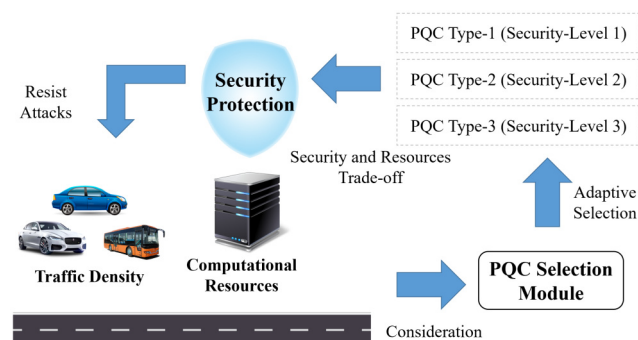


Fig. 1 Adaptive PQC parameter selection in ITS scenarios.

Quantum computing presents a direct threat to the cryptographic foundations securing ITS communications, given its

- Tongmao Ma and Shanchen Pang are with the School of Computer Science and Technology, China University of Petroleum, Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software, Qingdao 266580, China. E-mail: tongmao.ma@upc.edu.cn, pangsc@upc.edu.cn.
- Lu Liu is with the Department of Computer Science, University of Exeter, EX4 4PY Exeter, U.K. E-mail: l.liu3@exeter.ac.uk.
- Xiao Chen is with the School of Computing and Mathematical Sciences, University of Leicester, UK. E-mail: xiao.chen@leicester.ac.uk.
- Cong Liu is with NOVA Information Management School, Nova University of Lisbon, 1070-312 Lisbon, Portugal. E-mail: cliu@novaims.unl.pt.
- Tong Ding is with the School of Information Engineering, Ningxia University, China. E-mail: tongding@nxu.edu.cn.

Manuscript received: 2026-01-15; revised: 2026-03-08; accepted: 2026-03-20

potential to compromise classical public-key schemes such as RSA and ECC [6]. To mitigate this threat, the U.S. National Institute of Standards and Technology (NIST) has finalised its initial standardisation of Post-Quantum Cryptography (PQC), endorsing the lattice-based CRYSTALS-Kyber algorithm with security levels 512, 768, and 1024 as a key encapsulation mechanism [7, 8]. In this practical ITS deployments, however, link quality varies significantly due to fluctuating vehicle density and wireless interference, computational resources differ markedly between onboard and roadside units, and traffic patterns shift frequently between urban and highway contexts [9, 10]. Consequently, selecting a single, fixed Kyber parameter set cannot adequately balance security with latency and computational constraints: higher security levels provide strong quantum resistance but introduce substantial ciphertext overhead and delay, while lower levels reduce delay at the cost of diminished security assurance [11, 12].

Existing approaches for PQC parameter selection in ITS can be categorised broadly into offline static optimisation and online adaptive adjustment. Static optimisation methods typically determine optimal PQC parameters through preliminary benchmark testing or heuristic algorithms prior to system deployment [13, 14]. Their reliance on predefined scenarios, however, causes significant performance degradation in dynamic ITS environments where actual traffic loads and resource availability often diverge from initial assumptions. In contrast, online adaptive methods dynamically adjust Kyber parameters using rule-based strategies or deep reinforcement learning (DRL) techniques driven by real-time metrics [15, 16]. Despite their adaptability, current DRL-based online methods exhibit two principal limitations. Firstly, traditional DRL algorithms require continuous policy exploration during operation, leading to notable performance instability, particularly during abrupt changes in vehicle density or resource availability. Secondly, these online methods primarily select parameters based on immediate observations, lacking the capacity to anticipate forthcoming computational peaks or idle periods. This reactive, rather than proactive, characteristic restricts their ability to optimally reconcile security and efficiency within rapidly evolving ITS contexts.

To address these limitations, a novel predictive-adaptive framework, termed “**Predictive Deep Reinforcement Learning for Post-Quantum Cryptography Selection (PDRL-PQCS)**”, is proposed. The core concept of PDRL-PQCS is to integrate multi-step forecasting of traffic density and computational resources, facilitated by a Transformer-based predictor, into a DRL-driven adaptive strategy for PQC parameter selection. Utilising these predictive insights, the DRL agent proac-

tively selects from the CRYSTALS-Kyber security levels (512, 768, 1024), thereby effectively balancing cryptographic security with real-time performance demands. The principal contributions of this work are summarised as follows:

- A novel predictive-adaptive framework, PDRL-PQCS, is proposed, which constitutes a primary contribution by uniquely integrating a Transformer-based multi-step traffic and resource predictor with a DRL agent for cryptographic decision-making. This integration represents a significant methodological advancement, enabling proactive, context-aware parameter selection for post-quantum cryptography, which is a capability previously absent in related research. The predictive module facilitates anticipatory adaptation, allowing the system to optimise for forthcoming network states rather than merely reacting to present conditions.
- A bespoke RL strategy is engineered, featuring a custom reward function that formally quantifies the trade-off between cryptographic security and operational latency. Its principal innovation lies in synthesising real-time observations with predicted future states to inform its policy, permitting truly optimal real-time selection among the CRYSTALS-Kyber security levels. This approach transcends conventional reactive methods, directly addressing a critical gap in achieving both quantum resilience and system efficiency in edge computing-supported volatile ITS environments.
- A comprehensive evaluation under diverse ITS scenarios validates the novel PDRL-PQCS framework, demonstrating its capacity to address a critical gap in adaptive cryptographic management by achieving an approximate 10-20% performance improvement over baseline methods (e.g., traditional genetic algorithms, greedy strategies, and heuristic methods). Furthermore, ablation studies further substantiate the pivotal role of the integrated predictive module, confirming that its inclusion is essential for the enhanced system stability, adaptability, and resource efficiency delivered by this unique predictive-adaptive approach.

2 Related Work

This section presents studies on post-quantum cryptography and its application in intelligent transportation systems, and reinforcement learning methods for adaptive resource management in vehicular networks.

2.1 Post-Quantum Cryptography

Recent advancements in post-quantum cryptography are explored, including threat analyses, evaluation of candi-

date schemes, implementation techniques, and emerging applications [17, 18]. Bernstein et al. [5] analyzed the quantum threat to traditional encryption (RSA, ECC), exploring five candidate schemes based on codes, lattices, and other approaches, and provided insights into standardization and deployment challenges. Dey et al. [19] introduced an isogeny-based post-quantum signcryption scheme offering IND-CCA and EUF-CMA security with reduced communication costs. Nejatollahi et al. [13] reviewed lattice-based cryptography's progress, focusing on key technologies and their potential in public-key encryption and homomorphic encryption. Chamola et al. [6] assessed quantum computing's impact on 5G and network technologies, highlighting quantum-safe solutions like QKD. Alkim et al. [20] presented the quantum-safe digital signature scheme qTESLA, based on the Ring Learning with Errors (R-LWE) problem, showing strong security and efficiency. Canto et al. [21] proposed a polymorphic fault detection scheme for quantum-safe encryption, achieving efficient error detection with low performance overhead. Deshpande et al. [22] optimized a Verilog hardware implementation for HQC-based key encapsulation, improving performance on FPGA. Bidoux et al. [23] enhanced the RQC encryption scheme with new methods to reduce key sizes while boosting security, evaluating new combined attacks and complexity. Recent work on multi-domain time series generation, such as TimeDP [24], has explored domain-adaptive temporal modeling using prototype-based prompts, which offers insights for adapting prediction models to varying traffic scenarios. Advanced temporal modeling with Transformer architectures, such as rotary position embedding [25], offers improved handling of temporal dependencies and sequence prediction tasks.

2.2 PQC for Intelligent Transportation Systems

Recent studies have explored various strategies to secure intelligent transportation systems against quantum threats. Twardokus et al. [26] propose a machine learning-based spectrum optimization technique and a hybrid authentication protocol for vehicle-to-vehicle (V2V) communications. Chen et al. [27] introduce a hybrid certificate scheme combining post-quantum cryptography (PQC) and elliptic curve cryptography (ECC) to secure credential management in vehicle-to-everything (V2X) communications. Abdullah et al. [28] discuss the quantum risks to traditional cryptographic systems, analyze NIST-standardized PQC solutions, and highlight the need for lightweight and hybrid cryptographic approaches. Liu et al. [29] present LBRAKA, a quantum-resistant scheme for anonymous authentication and key agreement in VANETs,

ensuring strong privacy and public traceability. Aslam et al. [8] propose a secure communication framework for connected and autonomous vehicles combining quantum-resistant blockchain with Kyber-based PQC. Recent advances in spatiotemporal traffic forecasting, such as SSTZIP-GNN [30], have demonstrated the effectiveness of zero-inflated statistical models for handling sparse urban mobility data. For large-scale traffic forecasting, efficient Transformer architectures [31] have been proposed to address scalability challenges in modeling spatial dependencies. Furthermore, multimodal fusion frameworks [32] have been developed to integrate diverse data sources such as social media, real estate, and points of interest for comprehensive traffic situation forecasting.

2.3 Reinforcement Learning for Vehicular Resource Management

Recent studies have employed reinforcement learning (RL) to enhance resource management in vehicular networks, focusing on edge-computing-enabled task offloading, resource allocation, caching, and reliability [33–35]. Waqar et al. [36] propose MARL-based computation offloading for aerial-terrestrial vehicular networks in 6G. Ju et al. [37] apply DRL to jointly optimize secure offloading and resource allocation in vehicular edge computing systems, reducing delays and ensuring security. Aung et al. [38] develop DRL-powered VeSoNet for efficient, reliable content caching in vehicular social networks. Li et al. [39] introduce a DRL-driven collaborative framework minimizing service delays and failure costs in MEC-enabled vehicular networks. Atallah et al. [1] design an RL scheduler optimizing connected vehicle operations under dynamic traffic, while Ning et al. [3] employ deep Q-learning to optimize edge-node task offloading. Additionally, MARL approaches have effectively managed dense traffic, offering rapid convergence and robustness for spectrum sharing and offloading tasks [40–42]. Recent advances in MARL have explored structural coordination mechanisms to improve collaborative decision-making in complex multiagent systems [43]. For instance, MARL has been applied to optimize virtual coupled train operations in railway transit systems, demonstrating improved efficiency and safety through collaborative train control [44]. Federated graph learning frameworks such as FedMetro [45] have also been proposed for privacy-preserving spatiotemporal prediction in distributed transit systems. Beyond these model-free RL paradigms, another important direction is Model-Based Reinforcement Learning (MBRL) [46], where an agent learns a world model for planning or policy improvement. Unlike the reactive methods discussed above, MBRL enables anticipation

of future changes and has shown promise in sample-efficient domains [47]. However, the integration of predictive modeling with DRL for proactive decision-making in ITS security and cryptographic parameter selection remains unexplored. This gap motivates our proposed framework, which is detailed in Section 3.

3 Predictive DRL for Dynamic PQC Selection Scheme

This section introduces the PDRL-PQCS framework, providing a comprehensive overview of its methodology for adaptive PQC parameter selection tailored to dynamic ITS environments. It covers the research problem definition, Transformer-based prediction module, DRL-based adaptive PQC selection mechanism, and the integral training and deployment procedures.

3.1 Optimization Problem Formulation

It is considered that an ITS (shown in Fig. 1) is operated over a finite time horizon $t = 1, \dots, T$. At each decision epoch t , the system must choose one PQC algorithm or parameter set (e.g. Kyber-512, Kyber-768, Kyber-1024), indexed by $l \in L$. Each choice l is characterized by a security score σ_l (larger values signify stronger quantum resistance), a computational cost c_l , and an end-to-end latency δ_l that includes both encryption and transmission delay. Meanwhile, the available computing resource at time t is u_t . A binary decision variable is established with (1):

$$x_{t,l} = \begin{cases} 1, & \text{if parameter set } l \text{ is chosen at time } t, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

$$\forall t = 1, \dots, T, l \in L.$$

Then, this work seeks to maximize a weighted trade-off between security and latency over the horizon:

$$\max_{\{x_{t,l}\}} \sum_{t=1}^T \sum_{l \in L} (\sigma_l - \lambda \delta_l) x_{t,l}, \quad (2)$$

where $\lambda \geq 0$ balances the penalty for latency against the security benefit. At each epoch, exactly one PQC configuration must be selected:

$$\sum_{l \in L} x_{t,l} = 1, \quad \forall t = 1, \dots, T. \quad (3)$$

The chosen configuration must not exceed available compute resources:

$$\sum_{l \in L} c_l x_{t,l} \leq u_t, \quad \forall t = 1, \dots, T. \quad (4)$$

Finally, enforce binary decisions:

$$x_{t,l} \in \{0, 1\}, \quad \forall t = 1, \dots, T, l \in L. \quad (5)$$

Using the vector $x_t = (x_{t,l})_{l \in L}$, define $\Sigma = (\sigma_l)_{l \in L}$, $\Delta = (\delta_l)_{l \in L}$, and $c = (c_l)_{l \in L}$. Then the problem can be formulated with:

$$\begin{aligned} \max_{x_1, \dots, x_T} & \sum_{t=1}^T (\Sigma^\top x_t - \lambda \Delta^\top x_t) \\ \text{s.t.} & \mathbf{1}^\top x_t = 1, \quad c^\top x_t \leq u_t, \quad x_t \in \{0, 1\}^{|L|}, \quad \forall t. \end{aligned} \quad (6)$$

3.2 Transformer-based Predictive Module

As intelligent traffic systems grow, accurate forecasting of traffic flow and resource availability is essential for optimal resource allocation and ensuring the security of post-quantum algorithms. To address this, we propose a Transformer-based model to predict future traffic density and resource demands (shown in Fig. 2(a)). The model processes historical sequences with two-dimensional inputs (e.g., traffic flow, computational load) and uses positional encoding to embed temporal information. The Transformer encoder captures complex temporal dependencies, and the output layer predicts the next time step. More details are introduced as follows.

3.2.1 Model Structure Overview

The Transformer predictor consists of four main components: an input projection layer for embedding the input sequences, a positional encoding layer to add temporal information, multiple Transformer encoder layers using multi-head self-attention, and an output projection layer to generate the final predictions. This Transformer-based architecture is used to forecast future traffic density and resource requirements.

3.2.2 Input Projection

The initial step converts raw input data, consisting of historical traffic density and resource availability, into a continuous embedding space for richer, higher-dimensional processing. This transformation is typically a linear mapping:

$$X_{\text{embed}} = W_{\text{in}} X_{\text{input}} + b_{\text{in}} \quad (7)$$

where X_{input} is the raw input feature vector at a given time step, W_{in} is the learnable weight matrix, and b_{in} is the learnable bias vector for the input projection layer.

3.2.3 Positional Encoding

To enable the model to account for the temporal order of the input sequence (i.e., the sequence of past traffic and resource data), we add positional encodings to the input embeddings. Since the core Transformer architecture is permutation-invariant, explicit positional information is necessary. We utilize the standard sinusoidal positional encoding, which provides a unique vector for each position:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad (8)$$

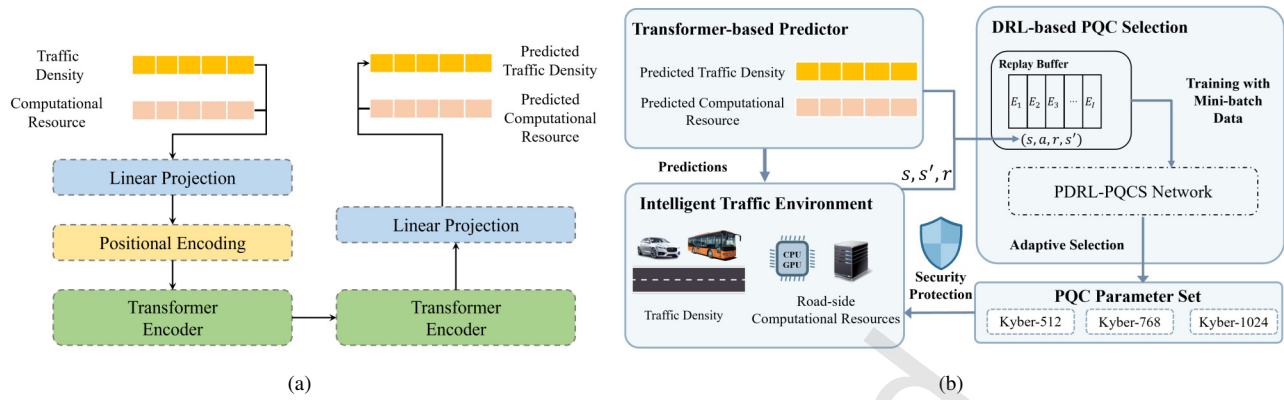


Fig. 2 Predicted module and RL-based learning framework for PDRL-PQCS. (a) Transformer-based traffic density and resource prediction model. (b) Reinforcement learning enhanced intelligent PQC selection for the traffic system.

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (9)$$

where pos is the position (time step index) within the sequence, i is the dimension index within the embedding vector, and d_{model} is the dimensionality of the model's embedding space. These positional encodings are added element-wise to the input embeddings X_{embed} .

3.2.4 Transformer Encoder Layer

The core processing of the sequence embeddings is performed by multiple stacked Transformer encoder layers. Each layer is responsible for processing the sequence through self-attention mechanisms, allowing the model to weigh the importance of different historical data points (traffic density and resources) when processing each element in the sequence. A standard encoder layer consists of two main sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Residual connections and layer normalization are applied around each sub-layer:

$$X' = \text{LayerNorm}(X + \text{MultiHeadAttention}(X, X, X)) \quad (10)$$

$$X'' = \text{LayerNorm}(X' + \text{FeedForward}(X')) \quad (11)$$

Here, X represents the input to the encoder layer (initially the combined input embedding and positional encoding, and the output of the previous layer for subsequent layers), X' is the output after the self-attention sub-layer, and X'' is the final output of the encoder layer after the feed-forward sub-layer.

3.2.5 Output Projection

Finally, the output representation from the last Transformer encoder layer is projected to the desired output space, which corresponds to the predicted values for traffic density and computational resources at the next time step. This final

projection is typically another linear transformation:

$$\hat{Y} = W_{out}X'' + b_{out} \quad (12)$$

where X'' is the output from the final encoder layer, W_{out} is the learnable weight matrix, b_{out} is the learnable bias vector for the output projection layer, and $\hat{Y}$ is the final predicted output vector.

3.3 Markov Decision Process Modeling

In this subsection, we introduce a DRL-based method (shown in Fig. 2(b)) to solve the dynamic PQC parameter selection problem in intelligent transportation systems. Firstly, we formalize this process as a Markov Decision Process (MDP), which provides a standard mathematical framework for making dynamic decisions about PQC algorithms in uncertain environments. An MDP is usually defined by the following five components (S, A, P, R, γ) , where S indicate the set of PQC agent's states, A represents the set of PQC agent's actions, P denotes the state transition probabilities, R is the reward function, and γ is the discount factor. The following contents describe each component in detail.

3.3.1 State Space (S)

The state space S encompasses all the relevant environmental information needed by the agent to make decisions. To enable the agent to select the optimal PQC parameters based on the current situation with a certain level of foresight, the state $s_t \in S$ at time step t is defined as a vector which is shown in (13). This vector not only includes the current real-time environmental metrics but also includes predictions of future environmental states:

Real-time Environmental Metrics:

- Current vehicle density ρ_t reflects the current congestion level on the road.

- Currently available computational resources u_t represent the computational capacity of the node performing cryptographic operations (e.g., an inverse measure of CPU and memory utilization).
- Current accumulated security risk ξ_t is used to quantify the potential security risk accumulated recently due to the selection of lower security level PQC parameters.

Predicted Environmental Metrics:

- Predicted vehicle density for the next k time steps $(\rho^{t+1}, \dots, \rho^{t+k})$.
- Predicted available computational resources for the next k time steps $(\hat{u}_{t+1}, \dots, \hat{u}_{t+k})$.

These predictions are provided by a dedicated time-series forecasting module defined previously. Including these predicted values in the state representation allows the agent to foresee potential network congestion or resource constraints, enabling more strategic decision-making.

$$s_t = [\rho_t, u_t, \xi_t, \hat{\rho}_{t+1}, \hat{u}_{t+1}, \dots, \hat{\rho}_{t+k}, \hat{u}_{t+k}]. \quad (13)$$

3.3.2 Action Space (A)

The action space A defines all the possible actions that the agent can take in each state. In this problem, the action space is discrete and finite, corresponding to the selectable parameter sets of the CRYSTALS-Kyber algorithm. These three actions correspond to selecting the PQC security levels 1, 3, and 5 defined by NIST. Each level has different trade-offs in terms of security, computational overhead, and communication latency. The task of the agent is to select an action $a_t \in A$ (shown in (14)) at each decision point t .

$$a_t \in \mathcal{A} = \{0, 1, 2\}. \quad (14)$$

3.3.3 Transition Probability (P)

The transition probability $P(s_{t+1} | s_t, a_t)$ defines the probability of transitioning to the next state s_{t+1} after executing action a_t in state s_t . In a complex ITS environment, this transition function is usually implicit, determined by the inherent dynamics of the environment (such as the natural evolution of traffic flow and changes in resource usage) as well as the impact of the chosen action a_t on the environment (for example, how the computational delay of different Kyber parameters affects subsequent resource states).

3.3.4 Reward Function (R)

At each time step t , the agent receives a reward R_t composed of an immediate performance term, a congestion penalty, and a rare attack penalty:

$$R_t = r_t^{(0)} - P_t - A_t. \quad (15)$$

- a) *Immediate Performance Reward*: The immediate reward $r_t^{(0)}$ balances security and latency:

$$r_t^{(0)} = \sigma(a_t) - \frac{\delta_{\text{net}}(\rho_t) + \delta_{\text{enc}}(a_t, u_t)}{10}, \quad (16)$$

$$\delta_{\text{net}}(\rho_t) = \delta_0 \left(1 + \left(\frac{\rho_t}{100}\right)^2\right), \quad \delta_0 = 10 \text{ ms}, \quad (17)$$

$$\delta_{\text{enc}}(a_t, u_t) = \frac{c(a_t)}{\max(u_t/100, 0.1)}, \quad c(a_t) \in \{5, 10, 20\}, \quad (18)$$

$$\sigma(a_t) \in \{0, 2, 4\} \text{ for actions } a_t \in \{0, 1, 2\}. \quad (19)$$

where, δ_{net} indicates the network latency and δ_{enc} is the encryption delay.

- b) *Congestion Penalty*: If ρ_t exceeds the density threshold τ , a penalty of magnitude p_m is applied for the next D steps. Let $\{p_t^{(j)}\}$ be the set of active penalty terms, then

$$P_t = \sum_j p_t^{(j)}. \quad (20)$$

- c) *Attack Penalty*: When the accumulated risk $\xi_t \geq 80$, an attack triggers with probability $\xi_t/100$, incurring

$$A_t = \begin{cases} 100, & \text{if attack occurs,} \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

In summary, the agent's reward at step t is

$$R_t = \sigma(a_t) - \frac{\delta_{\text{net}}(\rho_t) + \delta_{\text{enc}}(a_t, u_t)}{10} - \sum_j p_t^{(j)} - A_t. \quad (22)$$

3.4 Training Process of PDRL-PQCS

The training process of PDRL-PQCS occurs in two consecutive stages, which is shown in Algorithm 1. Each training episode spans the finite horizon T defined in the optimization problem of Section 3.1. In the first stage, the Transformer-based predictive module is trained to predict future traffic density and computational resource availability. In the second stage, the DQN agent is trained within a simulated traffic environment, utilizing the predictions made by the Transformer.

- a) *Transformer Predictor Training*: The Transformer predictor $f_{\text{trans}}(\cdot; \Theta_{\text{trans}})$ is trained to minimize the mean squared error (MSE) between predicted and actual next-step values. The training loss over a batch is defined as:

$$\mathcal{L}_{\text{trans}} = \frac{1}{B} \sum_{i=1}^B \|f_{\text{trans}}(X_i; \Theta_{\text{trans}}) - Y_i\|_2^2. \quad (23)$$

After convergence, the model parameters Θ_{trans} are saved for use during reinforcement learning.

- b) *Reinforcement Learning with DQN*: We instantiate the interactive environment and the deep Q network (DQN)

Algorithm 1 PDRL-PQCS: Predictive DRL for Dynamic PQC Selection

Inputs: Environment $\mathcal{E}$, Predictor f_{trans} , DQN parameters θ , replay memory D , batch size B_{dqn} , episodes N_{eps} , discount factor γ , exploration rate ϵ

- 1: Collect environment trajectories from $\mathcal{E}$, form dataset $\mathcal{D}_{\text{pred}} = \{(X_i, Y_i)\}$
- 2: Train Transformer predictor $f_{\text{trans}}(\cdot; \Theta_{\text{trans}})$ using $\mathcal{D}_{\text{pred}}$ by minimizing prediction loss
- 3: **for** $\text{episode} = 1$ to N_{eps} **do**
- 4: Initialize environment $\mathcal{E}$, obtain initial state s_0
- 5: Set step index $t \leftarrow 0$
- 6: **while** episode not terminated **do**
- 7: Obtain recent history X_t of length w
- 8: Predict next-step states $(\hat{\rho}_{t+1}, \hat{u}_{t+1})$
- 9: Construct augmented state s_t
- 10: Select action a_t using ϵ -greedy policy based on $Q(s_t, a; \theta)$
- 11: Execute action a_t in $\mathcal{E}$, observe next actual state s_{t+1} , reward r_t , and done flag d_t
- 12: Obtain updated history X_{t+1} , predict $(\hat{\rho}_{t+2}, \hat{u}_{t+2})$
- 13: Construct next augmented state s_{t+1}
- 14: Store transition (s_t, a_t, r_t, s_{t+1}) in D
- 15: **if** $|D| \geq B_{\text{dqn}}$ **then**
- 16: Sample mini-batch from D
- 17: Compute target values and update θ
- 18: **end if**
- 19: Set $t \leftarrow t + 1$
- 20: **end while**
- 21: **end for**
- 22: **return** trained DQN parameters θ

agent $Q(s, a; \theta)$, where θ are the Q-network parameters and θ^- are the target network parameters. Training proceeds over N_{eps} episodes. At the start of each episode, the environment is reset, yielding an initial actual state s_0^{act} and zero accumulated reward. At each time step t :

- **State Augmentation.** Retrieve the most recent history of length w from the environment, form the tensor X_t , and compute the next-step prediction

$$(\hat{\rho}_{t+1}, \hat{u}_{t+1}) = f_{\text{trans}}(X_t; \Theta_{\text{trans}}).$$

The augmented state is then

$$s_t = [\rho_t, u_t, \xi_t, \hat{\rho}_{t+1}, \hat{u}_{t+1}, \dots].$$

- **Action Selection.** Choose action a_t according to an

ϵ -greedy policy:

$$a_t = \begin{cases} \text{random action,} & \text{with probability } \epsilon, \\ \arg \max_a Q(s_t, a; \theta), & \text{otherwise.} \end{cases} \quad (24)$$

- **Environment Step.** Execute a_t , observe reward r_t , state s_t , and done flag d_t . Form the next augmented state s_{t+1} or set $s_{t+1} = \text{None}$. Store the transition (s_t, a_t, r_t, s_{t+1}) in the replay buffer.

- **Network Update.** Sample a minibatch of size B_{dqn} from the replay buffer and perform a gradient descent step on the loss:

$$\mathcal{L}_{\text{dqn}}(\theta) = \mathbb{E}_{(s, a, r, s')} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]. \quad (25)$$

Every C learning steps, the target network is updated via $\theta^- \leftarrow \theta$. The exploration rate decays at each step by factor τ until a floor $\epsilon_{\min}$.

This two-phase training process endows PDRL-PQCS with both accurate traffic and resource forecasts and robust dynamic decision-making for PQC parameter selection.

4 Experiment

To verify the effectiveness of the PDRL-PQCS method, this section designs experiments including environment setup, algorithm parameter settings, comparisons with baseline methods, and an ablation study on the Transformer module.

4.1 Experimental Environment Settings

4.1.1 Experimental Platform

Experiments were simulated using Python 3.8.5 and PyTorch 1.10.0, and the hardware environment consisted of a 3.70GHz Intel i9-10900K CPU (64 GB memory) and an NVIDIA RTX 3090 GPU for acceleration. In addition, a simulation-interaction framework was constructed on a Python 3.7 platform by means of NumPy and PyTorch.

4.1.2 Experimental Environment

The experimental setup simulates edge computing scenarios with fluctuating traffic demands and resource constraints. Traffic flow is modeled with a baseline value of 50, sinusoidal fluctuations (amplitude 20), Gaussian noise, and a congestion peak of +40 between steps 50-70. Similarly, resource availability follows a baseline of 80, with periodic fluctuations and a 20-unit reduction between steps 80-100, all values constrained within the range $[0, 100]$.



Table 1 Transformer Prediction Module Parameters

Parameter Name	Parameter Value
Input dimension	2
Model (embedding) dimension	32
Number of attention heads	4
Number of encoder layers	2
Dropout probability	0.1
Learning rate	0.001
Number of training epochs	2000

4.2 Experimental Parameter Configuration

4.2.1 Transformer Prediction Module

This work employs a Transformer model (parameters shown in Table 1) to process two-dimensional input sequences, where each time step comprises vehicle density and resource availability features. The input data are mapped into a 32-dimensional embedding space to capture temporal dependencies. The model adopts a 4-head parallel self-attention mechanism to identify critical information across different time steps. The encoder is constructed by stacking 2 identical layers, achieving a balance between representational capacity and computational cost. Dropout regularization with probability 0.1 is applied after each sub-layer to prevent overfitting. Training is carried out for 2000 epochs with a fixed learning rate of 0.001 for the optimizer, aiming to achieve accurate state prediction in complex dynamic environments.

4.2.2 RL-based Decision Module

In this work, the reinforcement learning decision module (parameters shown in Table 2) employs a 5-dimensional state vector, which includes current vehicle density, resource availability, cumulative risk, and sliding window historical data. The action space is defined as discrete encryption options with three levels: weak, medium, and strong. The Q-network adopts a single hidden layer structure with 64 neurons. Key training hyperparameters include a discount factor $\gamma = 0.99$ and a learning rate of 0.00005, ensuring training stability. An experience replay buffer with a capacity of 3000 transition samples is used, and 64 samples are drawn per iteration to update the network parameters. Exploration follows an ϵ -greedy policy: the initial exploration rate $\epsilon = 1.0$ decays progressively by a factor of 0.995 until reaching a lower bound of 0.05, achieving a gradual shift from extensive exploration to policy exploitation. After 20000 training rounds, the agent converges to the optimal encryption selection policy.

Table 2 DQN Decision Module Parameters

Parameter Name	Parameter Value
State dimension	5
Action dimension	3
Hidden layer dimension	64
Discount factor	0.99
Learning rate	0.00005
Replay buffer capacity	3000
Batch size	64
Initial epsilon for epsilon-greedy strategy	1.0
Minimum epsilon value	0.05
epsilon decay rate per step	0.995
Number of training episodes	20000

4.3 Evaluation Metric

To quantitatively evaluate the performance of our proposed method, we define the Traffic Security and Performance Index (TS-SPI), integrating security incentives, communication delays, congestion penalties, and attack penalties into a single metric. Specifically, the TS-SPI at each time step t is defined as:

$$\text{TS-SPI} = \sigma(a_t) - \frac{\delta_{\text{net}}(\rho_t) + \delta_{\text{enc}}(a_t, u_t)}{10} - P_{\text{delay}}(t) - P_{\text{attack}}(t), \quad (26)$$

where $\sigma(a_t)$ denotes the security incentive determined by encryption level a_t , $\delta_{\text{net}}(\rho_t)$ is the network delay related to vehicle density at time t , $\delta_{\text{enc}}(a_t, u_t)$ represents the encryption delay affected by resource availability, $P_{\text{delay}}(t)$ denotes the cumulative delay penalties triggered by congestion, and $P_{\text{attack}}(t)$ indicates the penalty due to security attacks. A higher TS-SPI value implies better overall system performance, indicating an effective balance among security robustness, computational efficiency, and traffic congestion management.

4.4 Baseline Methods for Comparison

To verify the effectiveness of the proposed PDRL-PQCS method, we select the following baseline algorithms for performance comparison:

- **Heuristic Policy:** A rule-based strategy employing pre-defined logic, such as selecting a fixed encryption level based on thresholds for resource availability or vehicle density.
- **Greedy Policy (Pure-Greedy):** A myopic approach that selects the action maximizing the immediate reward at each step, without considering long-term consequences.
- **Genetic Algorithm (Genetic-Alg):** An evolutionary computation approach potentially used to optimize

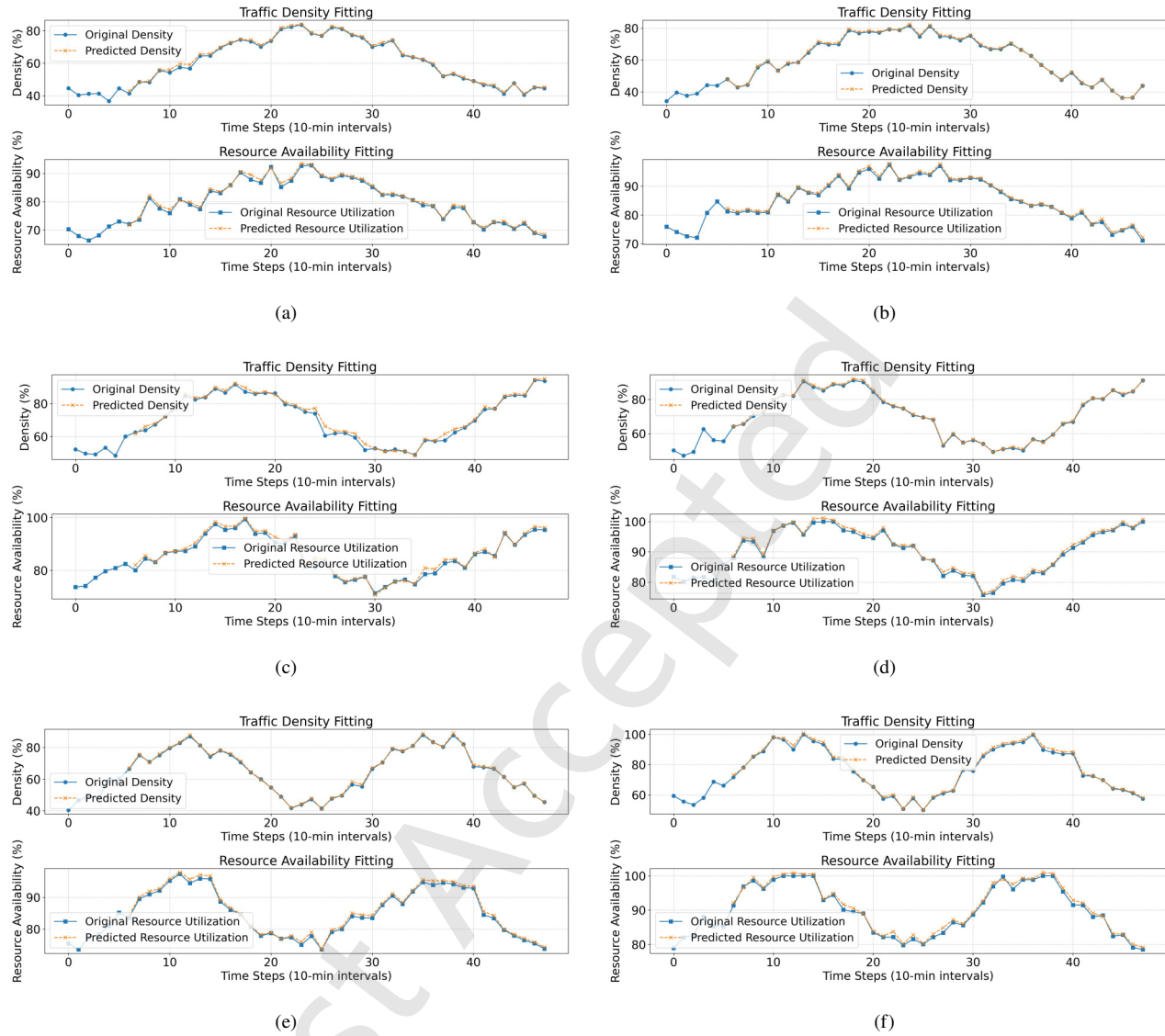


Fig. 3 Fitting performance of the transformer-based prediction model. (a) Traffic Type 1, $DS = 60$, $RS = 80$. (b) Traffic Type 1, $DS = 60$, $RS = 85$. (c) Traffic Type 2, $DS = 70$, $RS = 85$. (d) Traffic Type 2, $DS = 70$, $RS = 90$. (e) Traffic Type 3, $DS = 65$, $RS = 85$. (f) Traffic Type 3, $DS = 75$, $RS = 90$.

the decision-making policy through population-based search.

- RL-PQCS: A baseline implementation of the Deep Q-Network algorithm, excluding enhancements like the integrated Transformer prediction module present in our method.

4.5 Fitting Performance of the Prediction Model

In this work, a Transformer-based model is evaluated across three representative traffic scenarios, using two key indicators—vehicle density (DS) and resource availability (RS)—to thoroughly assess its adaptability to dynamic conditions. As illustrated in Fig. 3, the experimental results show that the

prediction model can capture the evolution of traffic states. Whether during periods of increasing density, peak congestion, recovery phases, or sharp fluctuations, the predicted trajectories remain consistent with the observed data, successfully reflecting both long-term trends and short-term variations. Similarly, resource availability predictions demonstrate high reliability, maintaining small prediction errors even during peak loads and sudden changes. Specifically, when $DS \leq 65$ and $RS \leq 85$, the predicted outcomes almost entirely overlap with the ground truth. Even under high-stress conditions ($DS \geq 70$, $RS \geq 90$), the model continues to deliver stable and accurate outputs. These results confirm the developed model's strong adaptability to complex traffic

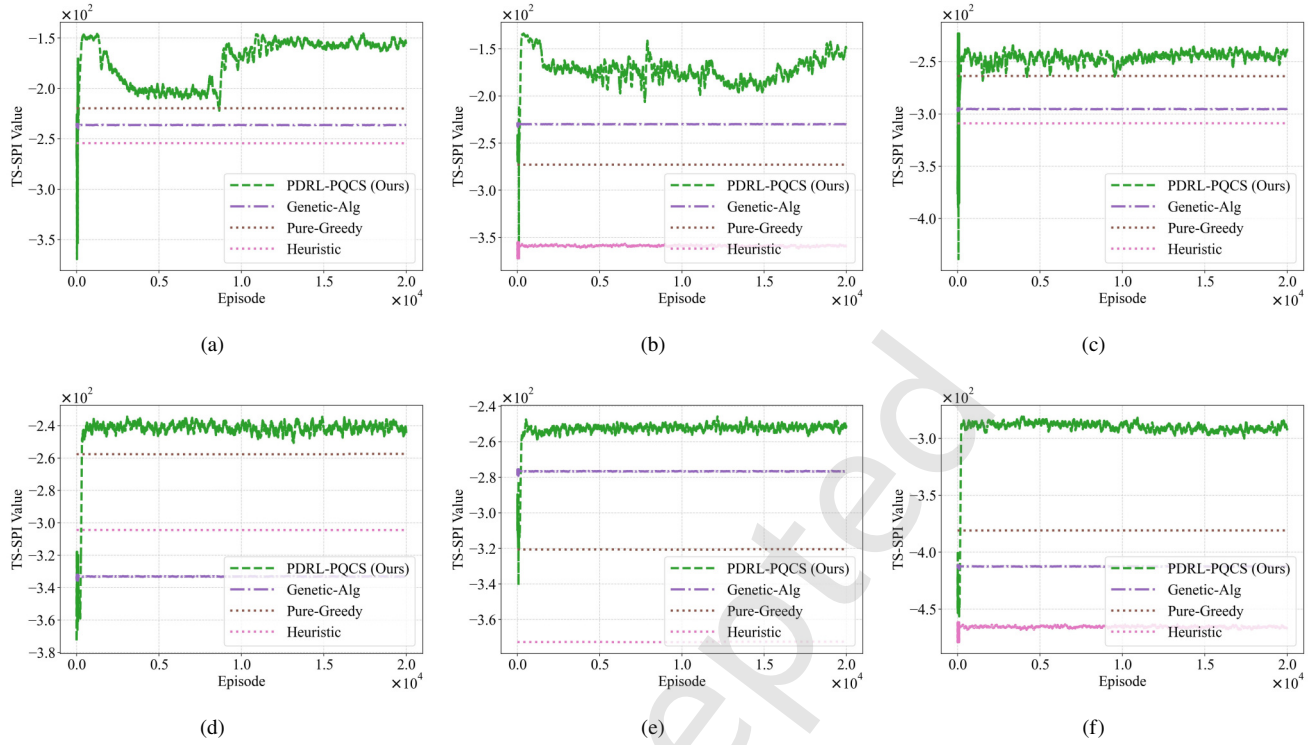


Fig. 4 Overall performance comparison of PDRL-PQCS and baseline methods. (a) Traffic Type 1, $DS = 60$, $RS = 80$. (b) Traffic Type 1, $DS = 60$, $RS = 85$. (c) Traffic Type 2, $DS = 70$, $RS = 85$. (d) Traffic Type 2, $DS = 70$, $RS = 90$. (e) Traffic Type 3, $DS = 65$, $RS = 85$. (f) Traffic Type 3, $DS = 75$, $RS = 90$.

environments and its outstanding generalization capability.

4.6 PQC Selection Performance Comparison

In this sub-section, the experiments evaluate the performance of three CRYSTALS-Kyber configurations (corresponding to NIST security levels 1/3/5) across various traffic types and different vehicle density and computational resource settings. As shown in Fig. 4, the developed model PDRL-PQCS outperforms baseline schemes across all tested scenarios. During the training process, the TS-SPI value increases rapidly and stabilizes at a higher level, indicating that the agent effectively learns the optimal encryption selection strategy. After approximately 5000 training rounds, the initial TS-SPI value fluctuations caused by exploration gradually disappear, and performance continues to improve. Under moderate constraint conditions in Type 1 scenarios (e.g., $DS = 60$, $RS = 80$ or 85), our method shows a clear advantage over genetic algorithms and heuristic methods. Although the pure greedy strategy demonstrates stable performance, it still falls short compared to our approach. Under harsher conditions (in Type 2 and Type 3 scenarios with $DS = 70$ or 75 and $RS = 90$), PDRL-PQCS maintains strong performance. When heuristic and greedy strategies fail under the high-stress Type 3 en-

vironment, our method quickly adapts and sustains a high TS-SPI value level—showing a 15%-20% improvement over heuristic methods and a 10% improvement over genetic algorithms. Even under extreme thresholds ($DS = 70$, $RS = 90$), the performance degradation remains minimal, validating its excellent generalization capability and stability. These results demonstrate that PDRL-PQCS can learn and maintain high-performance PQC selection policies across a range of traffic patterns and resource constraints, underscoring its practical robustness and versatility in intelligent transportation systems.

4.7 Ablation Study Analysis

We conducted an ablation study to assess the impact of the Transformer-based prediction module in our PQC selection framework. The full PDRL-PQCS framework was compared to a baseline RL-PQCS model that excludes the Transformer predictor, highlighting the benefits of the predictive component. The evaluation covers diverse scenarios, including three traffic types and varying vehicle density and resource utilization, to test the model's generalization capabilities. As presented in Fig. 5, the experimental results consistently show that PDRL-PQCS outperforms the baseline RL-PQCS across all tested traffic patterns and environmental settings.

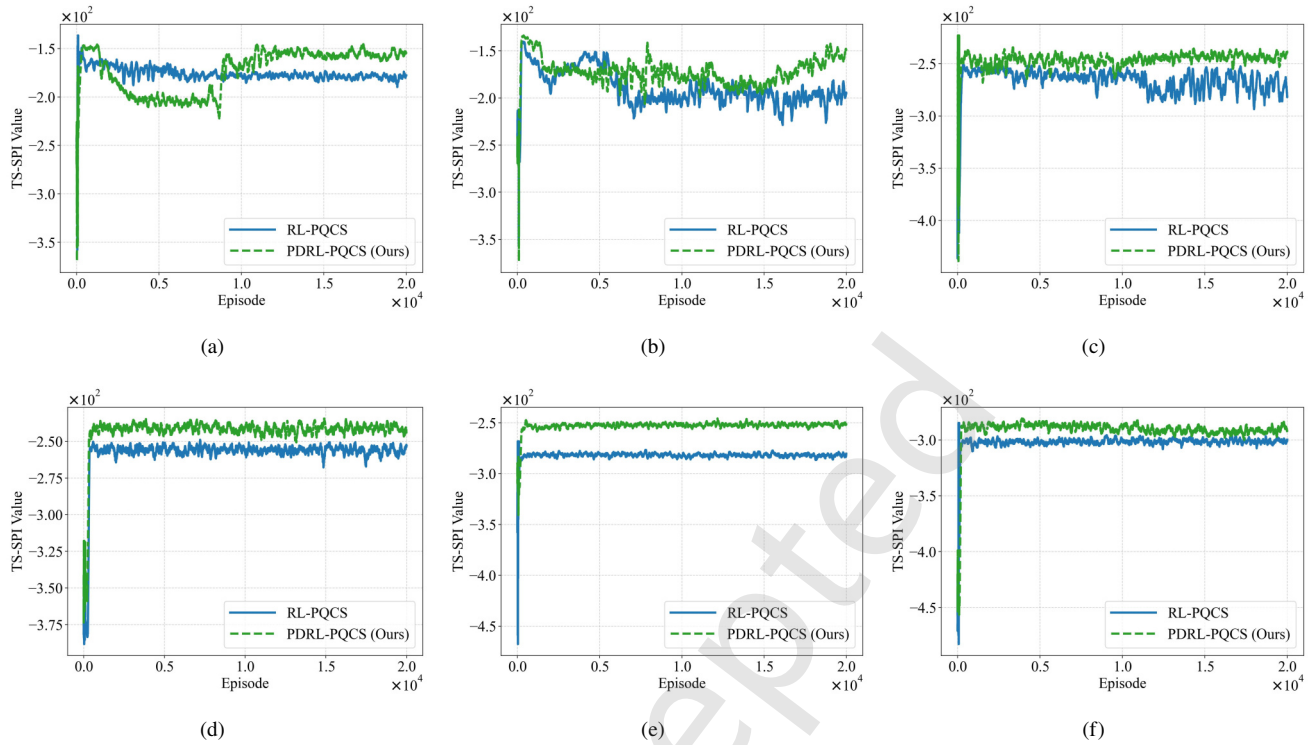


Fig. 5 Ablation comparison of the performance. (a) Traffic Type 1, $DS = 60$, $RS = 80$. (b) Traffic Type 1, $DS = 60$, $RS = 85$. (c) Traffic Type 2, $DS = 70$, $RS = 85$. (d) Traffic Type 2, $DS = 70$, $RS = 90$. (e) Traffic Type 3, $DS = 65$, $RS = 85$. (f) Traffic Type 3, $DS = 75$, $RS = 90$.

For instance, in Type 1 traffic scenarios characterized by lower vehicle density and ample resources (e.g., $DS = 60$, $RS = 80$), the two models initially achieve similar cumulative TS-SPI value. However, after about 4000 training iterations, PDRL-PQCS steadily surpasses RL-PQCS and ultimately converges to a higher and more stable performance level. When environmental constraints become tighter (e.g., DS raised to 65 or 70 and RS to 85 or 90), the superiority of PDRL-PQCS becomes even more pronounced: it maintains smoother learning curves and a higher average TS-SPI value, whereas RL-PQCS experiences larger fluctuations and slower convergence. In more challenging settings represented by Traffic Types 2 and 3, particularly under high-density and resource-constrained conditions (such as $DS = 75$, $RS = 90$), the advantage of PDRL-PQCS is especially evident. In these tough environments, RL-PQCS struggles to converge and hits a performance ceiling early, while PDRL-PQCS, benefiting from the Transformer module's ability to predict future states, achieves faster performance gains early in training and maintains superior TS-SPI value throughout. This advantage is reflected not only in the final performance but also in the overall training efficiency. The experimental results from the ablation study clearly demonstrate the crucial

role of the Transformer-based prediction module in enhancing the deep reinforcement learning framework for dynamic PQC selection. By enabling the agent to capture complex temporal patterns more effectively and anticipate environmental changes, the Transformer predictor significantly enhances the framework's adaptability, stability, and effectiveness across various traffic and environmental conditions.

5 Conclusion

This paper introduces the PDRL-PQCS framework, which is designed to enable adaptive selection of PQC parameters for ITS. The key innovation of the framework lies in the integration of a time-series forecasting module into the RL state representation, allowing it to predict dynamic variations in network congestion and computational resources. This capability enhances the framework's ability to make informed PQC decisions in complex and fluctuating environments. The experimental evaluation, conducted across three typical traffic patterns and varying vehicle density and resource availability settings, demonstrates that PDRL-PQCS outperforms baseline methods, achieving cumulative rewards 10% to 20% higher in challenging scenarios. Additionally, ablation studies validate the robustness of the Transformer-based prediction module, which plays a crucial role in ensuring long-term performance

stability.

Looking ahead, future work will extend the framework to multi-agent vehicular networks, where multiple roadside units and vehicles interact and make distributed decisions. This extension introduces core challenges such as non-stationarity, coordination among distributed agents, and partial observability, which will require new learning mechanisms and communication protocols. Alongside this direction, we will also explore its practical deployment through hardware-in-the-loop testing. Such testing will be essential to quantify the computational overhead introduced by running two neural networks online—measuring the actual latency and resource consumption of a prediction-decision cycle on representative automotive-grade platforms, and thereby validating whether the performance gains demonstrated in our simulations justify this additional overhead in practice.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62502539), the Natural Science Foundation of Shandong Province (Grant Nos. ZR2024MF129), and the national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS-<https://doi.org/10.54499/UID/04152/2025> (2025-01-01/2028-12-31) and UID/PRR/04152/2025 <https://doi.org/10.54499/UID/PRR/04152/2025> (2025-01-01/2026-06-30).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Reference

- [1] R. F. Atallah, C. M. Assi, and M. J. Khabbaz, "Scheduling the operation of a connected vehicular network using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1669–1682, 2018.
- [2] T. Ding, L. Liu, Z. Yan, X. Lu, and J. Hu, "Federated reinforcement learning for intelligent route planning in aerial-terrestrial network," *IEEE Internet of Things Journal*, pp. 1–1, 2024.
- [3] Z. Ning, P. Dong, X. Wang, J. J. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–24, 2019.
- [4] D. Xu, K. Yu, L. Liu, G. Chen, N. Kumar, M. Guizani, and J. A. Ritcey, "Post-quantum authentication against cyber-physical attacks in v2x-based autonomous vehicle platoon," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5034–5044, 2023.
- [5] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.
- [6] V. Chamola, A. Jolfaei, V. Chanana, P. Parashari, and V. Hassija, "Information security in the post quantum era for 5g and beyond networks: Threats to existing cryptography, and post-quantum cryptography," *Computer Communications*, vol. 176, pp. 99–118, 2021.
- [7] K. Zhang, M. S. Salek, A. Wang, M. Rahman, M. Chowdhury, and Y. Lao, "Preparing for kyber in securing intelligent transportation systems communications: A case study on fault-enabled chosen-ciphertext attack," 2025. [Online]. Available: <https://arxiv.org/abs/2502.01848>
- [8] A. M. Aslam, A. Bhardwaj, and R. Chaudhary, "Quantum-resilient blockchain-enabled secure communication framework for connected autonomous vehicles using post-quantum cryptography," *Vehicular Communications*, vol. 52, p. 100880, 2025.
- [9] Y. Gong and B. Hu, "A quantum-resistant key management scheme using blockchain in c-v2x," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [10] H. Gharavi, J. Granjal, and E. Monteiro, "Post-quantum blockchain security for the internet of things: Survey and research directions," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1748–1774, 2024.
- [11] R. Bavdekar, E. J. Chopde, A. Agrawal, A. Bhatia, and K. Tiwari, "Post quantum cryptography: a review of techniques, challenges and standardizations," in *2023 International Conference on Information Networking (ICOIN)*. IEEE, 2023, pp. 146–151.
- [12] S. Qiao, M. Huang, H. Li, L. Wang, W. Yin, Y. Sun, Z. Zhao, and Z. Lyu, "Fedssh: A consumer-oriented federated semi-supervised heterogeneous iomt framework," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 3, pp. 8465–8476, 2025.
- [13] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–41, 2019.
- [14] S. Qiao, Q. Guo, M. Wang, H. Zhu, J. J. P. C. Rodrigues, and Z. Lyu, "Frw-trace: Forensic-ready watermarking framework for tamper-resistant biometric data and attack traceability in consumer electronics," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 3, pp. 8234–8245, 2025.
- [15] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341–356, 2019.
- [16] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE*

- Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.
- [17] Z. Pei, C. Hong, and C. Wang, “Empowering np-hard problem solving in symmetric cipher attacks through the global optimization capability of the quantum tunneling effect,” *Tsinghua Science and Technology*, 2025.
- [18] Z. Pei, C. Hong, F. Xia, and C. Wang, “An innovative algorithm for attacking symmetric ciphers using d-wave quantum annealing,” *Tsinghua Science and Technology*, vol. 30, no. 5, pp. 2184–2194, 2025.
- [19] K. Dey, S. K. Debnath, P. Stănică, and V. Srivastava, “A post-quantum signcryption scheme using isogeny based cryptography,” *Journal of Information Security and Applications*, vol. 69, p. 103280, 2022.
- [20] E. Alkim, P. S. Barreto, N. Bindel, J. Krämer, P. Longa, and J. E. Ricardini, “The lattice-based digital signature scheme qtesla,” in *International Conference on Applied Cryptography and Network Security*. Springer, 2020, pp. 441–460.
- [21] A. C. Canto, A. Sarker, J. Kaur, M. M. Kermani, and R. Azarderakhsh, “Error detection schemes assessed on fpga for multipliers in lattice-based key encapsulation mechanisms in post-quantum cryptography,” *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, pp. 791–797, 2022.
- [22] S. Deshpande, C. Xu, M. Nawan, K. Nawaz, and J. Szefer, “Fast and efficient hardware implementation of hqc,” in *International Conference on Selected Areas in Cryptography*. Springer, 2023, pp. 297–321.
- [23] L. Bidoux, P. Briaud, M. Bros, and P. Gaborit, “Rqc revisited and more cryptanalysis for rank-based cryptography,” *IEEE Transactions on Information Theory*, vol. 70, no. 3, pp. 2271–2286, 2023.
- [24] Y.-H. Huang, C. Xu, Y. Wu, W.-J. Li, and J. Bian, “Timedp: Learning to generate multi-domain time series with domain prompts,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 17, 2025, pp. 17 520–17 527.
- [25] S. Dai, A. Gao, Z. Li, and Y. Du, “Rotary position embedding-based transformer hawkes process for event-type big data,” *Big Data Mining and Analytics*, vol. 9, no. 1, pp. 23–38, 2025.
- [26] G. Twardokus, N. Bindel, H. Rahbari, and S. McCarthy, “When cryptography needs a hand: Practical post-quantum authentication for v2v communications,” in *Network and Distributed System Security Symposium (NDSS 2024)*. Network and Distributed System Security Symposium (NDSS), 2024.
- [27] A. C. Chen and B.-Y. Lin, “Hybrid scheme of post-quantum cryptography and elliptic-curve cryptography for certificates-a case study of security credential management system in vehicle-to-everything communications,” in *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, vol. 1. IEEE, 2024, pp. 426–430.
- [28] A. A. Mamun, A. Abrar, M. Rahman, M. S. Salek, and M. Chowdhury, “Enhancing transportation cyber-physical systems security: A shift to post-quantum cryptography,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.13023>
- [29] G. Liu, W. Li, C. Yuan, N. Wang, C. Ma, N. Mu, Z. Liu, Y. Liu, and T. Xiang, “Lbraka: Lattice-based robust authenticated key agreement for vanets,” *IEEE Transactions on Vehicular Technology*, 2024.
- [30] Y. Shen, W. Shi, J. Shen, H. Wang, H. Wu, and J. Cao, “Multiresolution taxi demand prediction: A big data statistical and zero-inflated spatiotemporal gnn approach,” *Big Data Mining and Analytics*, vol. 9, no. 1, pp. 39–56, 2025.
- [31] Y. Fang, Y. Liang, B. Hui, Z. Shao, L. Deng, X. Liu, X. Jiang, and K. Zheng, “Efficient large-scale traffic forecasting with transformers: A spatial data management perspective,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2025, pp. 307–317.
- [32] B. Zhou, J. Liu, S. Cui, and Y. Zhao, “A large-scale spatiotemporal multimodal fusion framework for traffic prediction,” *Big Data Mining and Analytics*, vol. 7, no. 3, pp. 621–636, 2024.
- [33] T. Ding, L. Liu, Z. Yan, and L. Cui, “Enhancing sustainable data collection with unmanned aerial vehicles: A deep reinforcement learning approach,” *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 4657–4666, 2025.
- [34] T. Ding, N. Liu, Z. Yan, and L. Liu, Lei Cui, “An Efficient Reinforcement Learning Game Framework for UAV-Enabled Wireless Sensor Network Data Collection,” *Journal of Computer Science and Technology*, vol. 37, no. 6, pp. 1356–1368, 2022.
- [35] S. Qiao, H. Zhu, L. Sha, M. Wang, and Q. Guo, “Dynmark: A dynamic packet counting watermarking scheme for robust traffic tracing in network flows,” *Computers & Security*, vol. 157, p. 104571, 2025.
- [36] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, “Computation offloading and resource allocation in mec-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 478–21 491, 2022.
- [37] Y. Ju, Y. Chen, Z. Cao, L. Liu, Q. Pei, M. Xiao, K. Ota, M. Dong, and V. C. Leung, “Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5555–5569, 2023.
- [38] N. Aung, S. Dhelim, L. Chen, A. Lakas, W. Zhang, H. Ning, S. Chaib, and M. T. Kechadi, “Vesonet: Traffic-aware content caching for vehicular social networks using deep reinforcement learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8638–8649, 2023.
- [39] M. Li, J. Gao, L. Zhao, and X. Shen, “Deep reinforcement learning for collaborative edge computing in vehicular networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1122–1135, 2020.
- [40] H. Peng and X. Shen, “Multi-agent reinforcement learning based resource management in mec-and uav-assisted vehicular

networks,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 131–141, 2020.

- [41] J. Wang, B. Wang, S. Zhou, B. Cao, W. Li, and P. Zheng, “Dnacse: Enhancing genomic llms with contrastive learning for dna barcode identification,” *Journal of Chemical Information and Modeling*, vol. 66, no. 2, pp. 976–993, 2026.
- [42] S. Qiao, Q. Guo, M. Wang, H. Zhu, J. J. Rodrigues, and Z. Lyu, “Advances in network flow watermarking: A survey,” *Computers & Security*, vol. 159, p. 104653, 2025.
- [43] Y. Li, Y. Huang, J. Feng, C. Deng, C. Liu, V. Chau, and W. Wang, “Multiagent reinforcement learning based on structural coordination,” *Tsinghua Science and Technology*, 2025.
- [44] G. Jing, Y. Zou, Z. Zhang, J. Yu, D. Yu, and X. Cheng, “Multi-agent reinforcement learning optimization for virtual coupled train set problem in cbtc systems,” *Tsinghua Science and Technology*, 2025.
- [45] T. Zhang, X. He, Y. Wang, Y. Xu, R. Wu, Z. Wang, and Y. Tong, “Fedmetro: Efficient metro passenger flow prediction via federated graph learning,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5215–5224.
- [46] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, and Y. Yu, “A survey on model-based reinforcement learning,” *Science China Information Sciences*, vol. 67, no. 2, p. 121101, 2024.
- [47] J. Bayrooti, C. H. Ek, and A. Prorok, “Efficient model-based reinforcement learning through optimistic thompson sampling,” *arXiv preprint arXiv:2410.04988*, 2024.

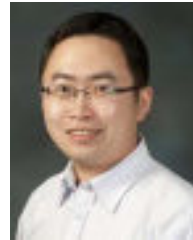
Author biography



Tongmao Ma obtained the PhD degree at Departamento de Inteligencia Artificial, ET-SIINF, Universidad Politécnica de Madrid, Spain. He received M.E. and B.S. degrees from China University of Petroleum (East China). He is currently employed as a post-doctoral researcher at China University of Petroleum (East China). His research interests include edge computing and DNA neural networks.



Shanchen Pang received the graduation degree from the Tongji University of Computer Software and Theory, Shanghai, China, in 2008. He is a Professor of the School of Computer Science and Technology of the China University of Petroleum, Qingdao, China. He has published more than 150 papers in international journals (including IEEE TII, TMC, IOTJ, etc). His current research interests include cloud computing, artificial intelligence, Petri net and reinforcement learning.



Lu Liu received the PhD degree from Surrey Space Centre, the University of Surrey. He had worked as a research fellow with the WRG eScience Centre, University of Leeds. He served as the head of the School of Computing and Mathematical Sciences, University of Leicester, U.K., from 2019–2023. His research interests are in the areas of data analytics, service computing, sustainable computing, and the Internet of Things. He has more than 250 scientific publications in reputable journals, academic books, and international conferences. He has been the recipient of 7 Best Paper Awards from international conferences and was invited to deliver 8 keynote speeches with international conferences. He is a fellow of BCS (British Computer Society). He is currently serving as an associate editor for Peer-to-Peer Networking and Application (PPNA) and Big Data Mining and Analytics (BDMA).



Xiao Chen holds MSc and PhD degrees in computing science from Newcastle University (UK) and currently serves as a lecturer in the School of Computing and Mathematical Sciences at the University of Leicester. Previously, he was a Marie Skłodowska-Curie fellow at the School of Informatics at the University of Edinburgh (UK). His research interests revolve around the theory of distributed systems, blockchains, and consensus protocols, with a focus on blockchain-supported distributed AI systems. Additionally, he specialises in stochastic and formal modelling, as well as AI-driven optimisation for large-scale systems.



Cong Liu received the BS and the MS degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2013 and 2015, respectively. He received his PhD degree in the Department of Mathematics and Computer Science at Eindhoven University of Technology in 2019. He is an invited full professor in the NOVA Information Management School, Nova University of Lisbon. His research interests are in the areas of process mining, business process management, and artificial intelligence.



Tong Ding is a lecturer in the School of Information Engineering at Ningxia University, China. He received his Ph.D. degree in Artificial Intelligence from Shandong University, China, in 2025, and M.Eng. and B.Eng. degrees in software engineering from Nanjing University of Posts and Telecommunications, Nanjing, and China University of Petroleum, Qingdao, in 2016 and 2020, respectively. His research interests include edge-cloud computing, applied machine learning, and drones' coordination.