

Utilisation Industrielle de TD(0), TD(n), Monte Carlo et SARSA : Fraude, Cybersécurité, Data Centers, Trading, Robotique, Recommandation

1 Introduction

Les méthodes TD(0), TD(n), Monte Carlo et SARSA sont souvent décrites dans des contextes académiques abstraits. Pourtant, elles possèdent des applications **réelles et massives** dans l'industrie : fraude bancaire, cybersécurité, conduite autonome, optimisation de data centers, recommandation, maintenance prédictive, trading algorithmique, robotique, etc.

Ce document présente une explication détaillée et concrète de ces méthodes avec :

- des scénarios industriels réels,
- la logique derrière chaque méthode,
- les avantages et limites dans la pratique,
- une section dédiée à SARSA et aux politiques prudentes,
- un tableau comparatif final pour l'ingénierie.

L'objectif est d'expliquer **quand** et **pourquoi** utiliser : TD(0), TD(1–3), TD(n), Monte Carlo ou SARSA dans une application réelle.

2 Détection de fraude (banque, paiements)

2.1 TD(0) : décision instantanée

Dans les systèmes anti-fraude réels, la plupart des décisions critiques doivent être prises immédiatement. Le modèle reçoit un événement (paiement, login, géolocalisation) et doit mettre à jour l'estimation du risque et l'action (en bloquant ou non) en une seule étape.

Exemples :

- Transaction de 3000\$ à Tokyo alors que l'utilisateur est à Montréal.
- Paiement venant d'une IP jamais vue dans un pays extrêmement risqué.

Dans ces cas :

- une mise à jour après un seul signal est suffisante,
- l'urgence prime sur la précision globale,
- une action doit être prise dès la première anomalie.

C'est exactement la logique de TD(0).

2.2 TD(1–3) : corrélation courte de signaux

Certaines fraudes se détectent avec deux ou trois événements successifs. Exemples :

- 1^{re} transaction suspecte : flag.
- 2^e transaction suspecte : risque accru.

- 3e : blocage définitif ou double authentification.

Le système utilise donc un retour sur 2–3 événements avant de corriger sa valeur de risque, ce qui correspond à TD(1)–TD(3).

2.3 TD(n) : modèles horizon court

Les modèles industriels de scoring utilisent des fenêtres temporelles : 5 minutes, 10 minutes, 30 minutes. Ils évaluent la séquence des événements sur une fenêtre, puis appliquent une mise à jour.

Exemples :

- attaques coordonnées multi-comptes sur une courte période,
- achats inhabituels répartis sur 15 minutes.

Ce fonctionnement correspond à TD(n) avec n plus grand.

2.4 Monte Carlo : analyse de fraude confirmée

Les équipes d'audit et de conformité utilisent une logique Monte Carlo :

- analyse complète de tout l'historique client,
- reconstruction des séquences,
- décision finale “fraude avérée” après toute la séquence.

3 Cybersécurité : attaques, IDS/IPS, forensic

3.1 TD(0) : IDS/IPS (Snort, Suricata, WAF)

Les systèmes d'intrusion en temps réel prennent une décision sur un seul événement.

Exemples :

- un paquet contenant un exploit blocage immédiat,
- 100 tentatives SSH en 5 secondes bannissement instantané.

Une seule observation suffit à mettre à jour la croyance sur l'état du système. TD(0) est indispensable dans la détection en temps réel.

3.2 TD(1–3) : corrélation légère

Exemple d'un use-case Splunk ou Sentinel :

1. scan des ports,
2. brute force SSH,
3. tentative sur /wp-login.php.

Action typique :

- log au premier,
- alerte au deuxième,
- blocage au troisième.

3.3 TD(n) : kill-chain MITRE ATT&CK

Certaines intrusions suivent une séquence longue :

1. Reconnaissance,
2. Phishing,

3. Accès VPN volé,
4. Mouvement latéral,
5. Exfiltration.

Les systèmes SIEM avancés détectent une attaque **après n événements significatifs**. C'est du TD(n) réel.

3.4 Monte Carlo : forensic

En post-incident :

- on analyse la séquence complète de logs,
- on reconstruit la timeline,
- on conclut l'incident uniquement **après** avoir tout vu.

4 Optimisation de data centers (Google, Microsoft)

4.1 TD(0) : autoscaling immédiat

Dans Kubernetes HPA :

- si CPU > seuil ajouter un pod,
- si CPU < seuil retirer un pod.

Mise à jour basée uniquement sur l'état actuel + estimation suivante.

4.2 TD(1–3) : prévision courte

Google Borg RL :

- prédiction à 1–3 minutes,
- éviter les oscillations de scaling.

4.3 TD(n) : horizon plus long

Optimisation énergétique à horizon 10 minutes : mise à jour de la stratégie après une séquence plus longue.

4.4 Monte Carlo : analyse de jobs HPC

En fin de tâche :

- calcul total d'énergie,
- analyse complète des performances.

5 Trading algorithmique

5.1 TD(0)

Algorithmes à la milliseconde :

- décision basée sur le prochain “tick”,
- mise à jour immédiate du modèle.

5.2 TD(1–3)

Stratégies à horizon court :

- prédiction directionnelle sur quelques ticks,
- arbitrage court terme.

5.3 TD(n)

Prévision à horizon 10–30 ticks : micro-tendances.

5.4 Monte Carlo

Backtesting complet :

- stratégie testée sur 10 ans de données,
- décision finale basée sur retour complet.

6 Robotique industrielle et conduite autonome

6.1 TD(0)

Contrôle continu :

- ajustement immédiat de forces/torques,
- collision avoidance instantané.

6.2 TD(1–3)

Prédiction de trajectoire à court terme :

- 1–3 frames pour anticiper obstacles,
- stabilisation d'un bras robotisé.

6.3 TD(n)

Horizon 10–20 frames :

- planification moyenne (6–12 m),
- anticiper mouvements piétons/voitures.

6.4 Monte Carlo

Simulation complète d'un trajet :

- évaluation totale,
- métriques de sécurité globales.

7 Recommandation (YouTube, TikTok, Netflix)

7.1 TD(0)

Mise à jour du score dès :

- un clic,
- un like,
- 1 seconde de visionnage.

7.2 TD(1–3)

Rétention à 3 secondes :

- prédiction si l'utilisateur continue de regarder.

7.3 TD(n)

Watch-time complet (mais partiel) :

- prédiction horizon 10–30 secondes.

7.4 Monte Carlo

Détermination du watch-time réel une fois la vidéo terminée.

8 Le rôle de SARSA : politiques prudentes

SARSA est on-policy : il met à jour les valeurs selon l'action réellement prise. Il est utilisé dans les environnements où la **prudence** est essentielle.

8.1 Exemples industriels

Cybersécurité Un agent d'auto-blocage apprend sa politique réelle :

- éviter les faux positifs,
- rester prudent,
- ne pas verrouiller trop agressivement.

Robotique Un robot collaboratif (cobot) doit être sûr :

- SARSA évite d'apprendre des politiques dangereuses,
- car il suit ce qu'il fait vraiment (safe moves).

Finance Stratégies prudentes (risque limité) :

- SARSA apprend l'action réellement prise sous risque limité,
- contrairement à Q-Learning qui tend vers une stratégie agressive.

9 Tableau comparatif final

Méthode	Utilisation industrielle typique
TD(0)	Décision immédiate : fraude en ligne, IDS/IPS, autoscaling, trading microsecondes, robotique instantanée.
TD(1–3)	Prévision courte : analyse de signaux successifs, trajectoires courtes, corrélation légère, rétention courte.
TD(n)	Horizon moyen : kill-chain cyber, maintenance prédictive fenêtre 10s, trading 10–30 ticks, recommandation 30s.
Monte Carlo	Analyse complète : forensic, audit, backtesting, watch-time complet, simulation d'un trajet entier.
SARSA	Politiques prudentes : cybersécurité sensible, robotique collaborative, stratégies à risque limité.
Q-Learning	Politiques optimales agressives : trading opportuniste, agents compétitifs, optimisation extrême.

10 Conclusion

Les méthodes TD(0), TD(n), Monte Carlo et SARSA forment un ensemble d'approches complémentaires permettant de modéliser des systèmes industriels complexes. Le choix dépend du degré de réactivité nécessaire, de la longueur de la séquence à observer, et du niveau de prudence souhaité. SARSA apporte une dimension spécifique adaptée aux environnements où la sécurité ou la stabilité sont cruciales.