



RAPORT DE ACTIVITATE

Titlu proiect: Tehnici digitale pentru controlul și optimizarea traficului rutier prin utilizarea de algoritmi de inteligență artificială

Etapa: 4 (1.07.2024 – 4.12.2024)

Obiectivele etapei sunt :

1. Validarea în simulare și experimentarea metodelor prezentate la etapele precedente - tehnici de inteligență artificială pentru controlul timpilor de semaforizare în funcție de încărcarea segmentelor de drum în scopul reducerii timpilor de așteptare ;
2. Evidențierea câștigurilor obținute și a posibilităților de implementare

1. Congestia rutieră

Pentru a defini congestia rutieră putem pleca de la reprezentarea bazată pe date experimentale din Figura 1 [1], unde este reprezentată dependența fluxului rutier(q) de densitatea rutieră(ρ).

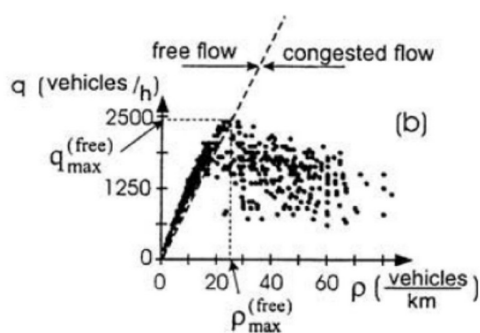


Figura 1. Diagramă fundamentală de trafic

Congestia rutieră poate astfel fi interpretată ca fenomenul care apare atunci când valorile crescute de trafic determină depășirea valorii critice a densității rutiere.

2. Soluții pentru reducerea congestiei

Principalele metode pentru reducerea congestiei sunt :

- modificarea infrastructurii ;
- îmbunătățirea transportului public/alternativ ;
- controlul traficului prin utilizarea semaforizării inteligente.

Prima metodă este destul de costisitoare și implică discomfort crescut pentru participanții la trafic. Cea de-a doua metodă depinde de disponibilitatea autorităților locale de a aloca fonduri pentru achiziția de noi unități pentru a mări flota disponibilă pentru transportul în comun, precum și pentru reabilitarea infrastructurii existente. Cea de-a treia metodă constituie obiectivul cercetărilor acestui proiect, întrucât presupune îmbunătățirea condițiilor de trafic fără a perturba într-o manieră semnificativă infrastructura existentă.

3. Controlul traficului - semaforizarea inteligentă

Controlul traficului se referă la utilizarea unor algoritmi care să calculeze eficient timpul de semaforizare pentru a putea gestiona mai bine situația traficului la orele de vârf. Pentru a putea controla traficul, am utilizat două tehnici de inteligență artificială din cadrul Machine Learning, după cum poate fi văzut în Figura 2.

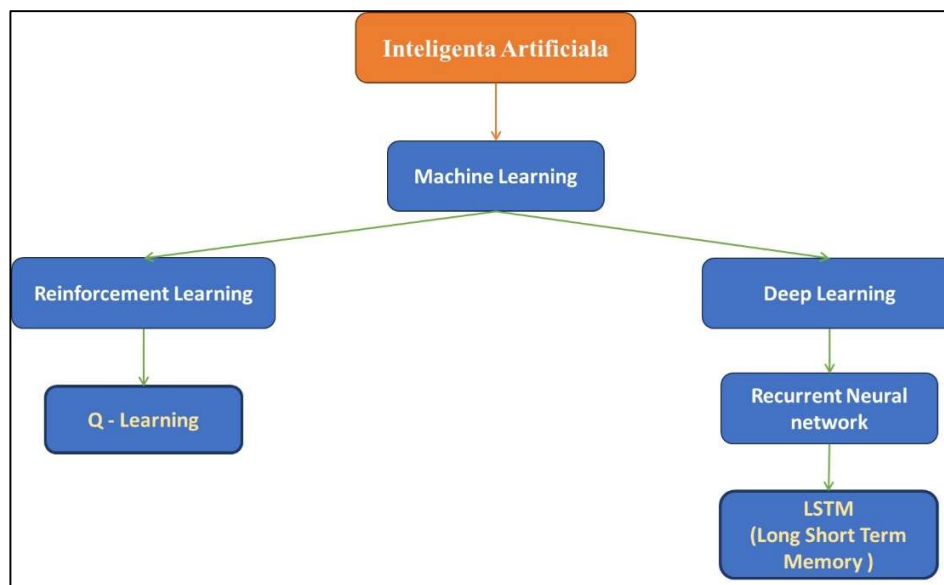


Figura 2. Tehnici de inteligență artificială utilizate.

3.1. Q – Learning

Primul algoritm utilizat, Q-Learning, face parte din Machine Learning, clasa Reinforcement Learning (RL - învățarea prin consolidare). Algoritmul a fost introdus pentru prima dată în anul 1989 de Watkins [2]. RL este un domeniu al învățării automate care se bazează pe modul în care agenții ar trebui să decidă cum să acționeze într-un anumit mediu pentru a maximiza noțiunea de recompensă cumulativă. Acesta devine un instrument major în inteligența artificială și tehnica computațională, în care computerele își fac propriile alegeri într-un mediu dat fără a avea nicio idee despre date istorice sau etichetate. [3, 4]

Q-Learning utilizează un agent care învață funcția valorii pentru o anumită politică prin interacțiunea cu mediul pentru a prezice o soluție optimă. Pe baza funcției valorii, dezvoltă și învață continuu politica optimă [5].

Scopul Q-learning este de a învăța calitatea acțiunilor, care este reprezentată de valorile Q. Aceste valori Q sunt utilizate pentru a determina cea mai bună acțiune de luat în orice stare dată și are un avantaj foarte mare, acela că nu necesită o rețea neuronală, ci folosește o tabelă (Q) pentru a găsi cea mai bună acțiune posibilă de întreprins într-o anumită situație. Q se referă la funcția pe care o calculează algoritmul, adică recompensele așteptate pentru o acțiune întreprinsă, într-o stare dată. Algoritmul se bazează pe o tabelă Q, care este compusă din stări (valori ale traficului) și acțiuni (timpii de semaforizare). Starea curentă se actualizează la fiecare pas pe baza stării anterioare și recompensei calculate pentru situația respectivă.

$$Q(S_t^1(i), S_t^2(j)) = Q(S_{t-1}^1(i), S_{t-1}^2(j)) + \text{sign} \left(\beta \cdot \frac{n_{last}(t-1)}{n_{last}(t-2)} + \gamma \cdot \frac{n_{acc}(t-1)}{n_{acc}(t-2)} \right) \quad (1)$$

Stare curentă
Stare anterioară
recompensă

Q-learning găsește o politică optimă pentru a maximiza valoarea așteptată a recompensei totale pe parcursul tuturor etapelor succesive.

3.2 Deep – Learning

Al doilea algoritm face parte din clasa Deep Learning, subclasa Rețele neurale recurente, arhitectura Long Short-Term Memory (LSTM). RNN și LSTM sunt arhitecturi speciale de rețele neuronale care sunt capabile să proceseze date secvențiale, date în care ordonarea cronologică contează. LSTM-urile sunt versiuni în esență îmbunătățite ale RNN-urilor, capabile să interpreteze secvențe mai lungi de date. Modelele LSTM sunt compuse din trei componente diferite, sau porți. Există o poartă de intrare, o poartă de ieșire și o poartă de uitare. La fel ca RNN, LSTM-urile iau în considerare intrările din pasul de timp anterior atunci când modifică memoria modelului și greutatea de intrare. Poarta de intrare ia decizii cu privire la valorile importante și ar trebui lăsate să treacă prin model. O funcție sigmoidă este utilizată în poarta de intrare, care face determinări despre ce valori să se transmită prin rețeaua recurentă. Zero scade valoarea, în timp ce 1 o păstrează. Aici este folosită și o funcție TanH, care decide cât de importante sunt pentru model valorile de intrare, variind de la -1 la 1.

După ce intrările curente și starea memoriei sunt luate în considerare, poarta de ieșire decide ce valori să trimită la următorul pas de timp. În poarta de ieșire, valorile sunt analizate și atribuite o importanță variind de la -1 la 1. Aceasta reglează datele înainte de a fi trecute la următorul calcul al pasului de timp. În cele din urmă, sarcina porții de uitare este să neglijeze informațiile pe care modelul le consideră inutile pentru a lua o decizie cu privire la natura valorilor de intrare. Poarta de uitare folosește o funcție sigmoidă asupra valorilor, scoțând numere între 0 (uitați asta) și 1 (păstrați asta). [6]

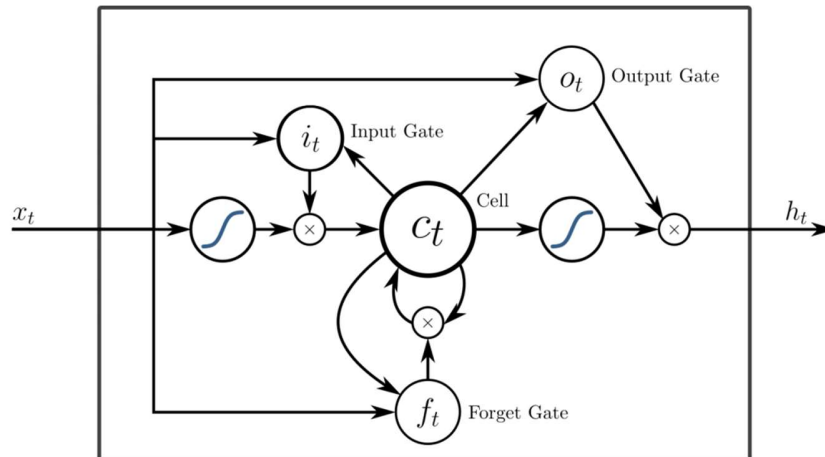


Figura 3. Rețele neurale recurente.

Pentru a putea folosi o rețea de tipul LSTM, am construit o arhitectură bazată pe acest algoritm, descris în Figura 4. Rețeaua clasifică situațiile din trafic în două stări : ”Crește timp verde” și ”Scade timp verde”. În funcție de aceste stări, sistemul va lua deciziile de modificare a timpului de semaforizare.

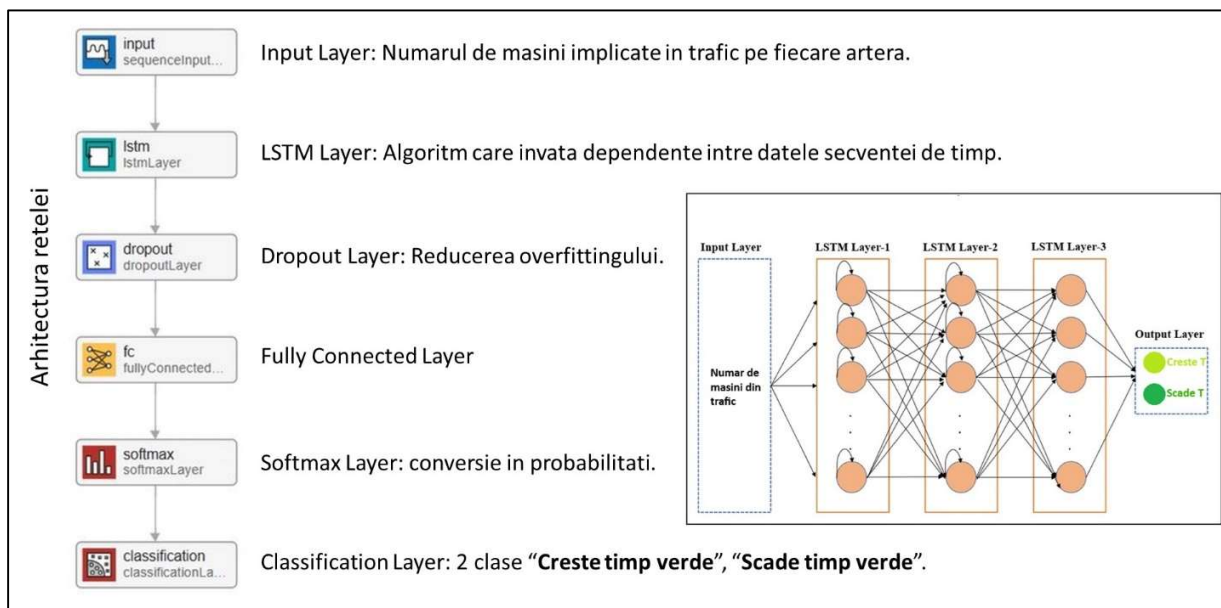
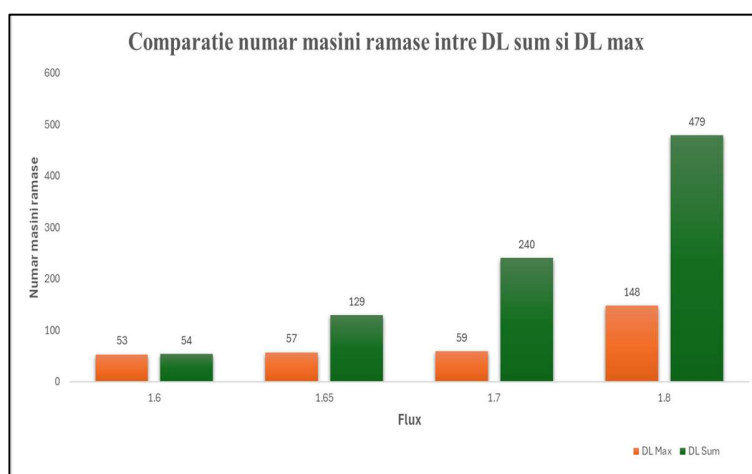


Figura 4. Arhitectura rețelei LSTM.

Rețeaua a fost antrenată pe date de trafic reale [7]. Au fost utilizate două modalități diferite pentru alegerea etichetelor “Crește timp verde”, “Scade timp verde”.

Indicatorii folosiți au fost numărul maxim de mașini din trafic: rețeaua DLmax și numărul total de mașini implicate în trafic: rețeaua DLsum. Numarul de mașini rămase reprezintă mașinile care nu trec în primul ciclu de semaforizare și aglomerează intersecția. Rețeaua DLmax a dat rezultate net superioare celei DLsum, după cum se poate vedea în Figura 5.

Figura 5. Comparatie retea DL max și DL sum.



3.3 Rezultate semaforizare cu timp fixat

Dupa cum se poate observa în Figura 6, în cazul semaforizării cu timp fix, pe durata traficului de vârf se formează aglomerație (graficul galben) datorită numarului de mașini care nu reușesc să treacă în primul ciclu de semaforizare. Semaforul are valoare maximă, iar numărul de mașini acumulate va fi proporțional cu acest timp, de aceea la perioada de vârf acumularea va depăși capacitatea albastră de trecere și se va forma congestie.

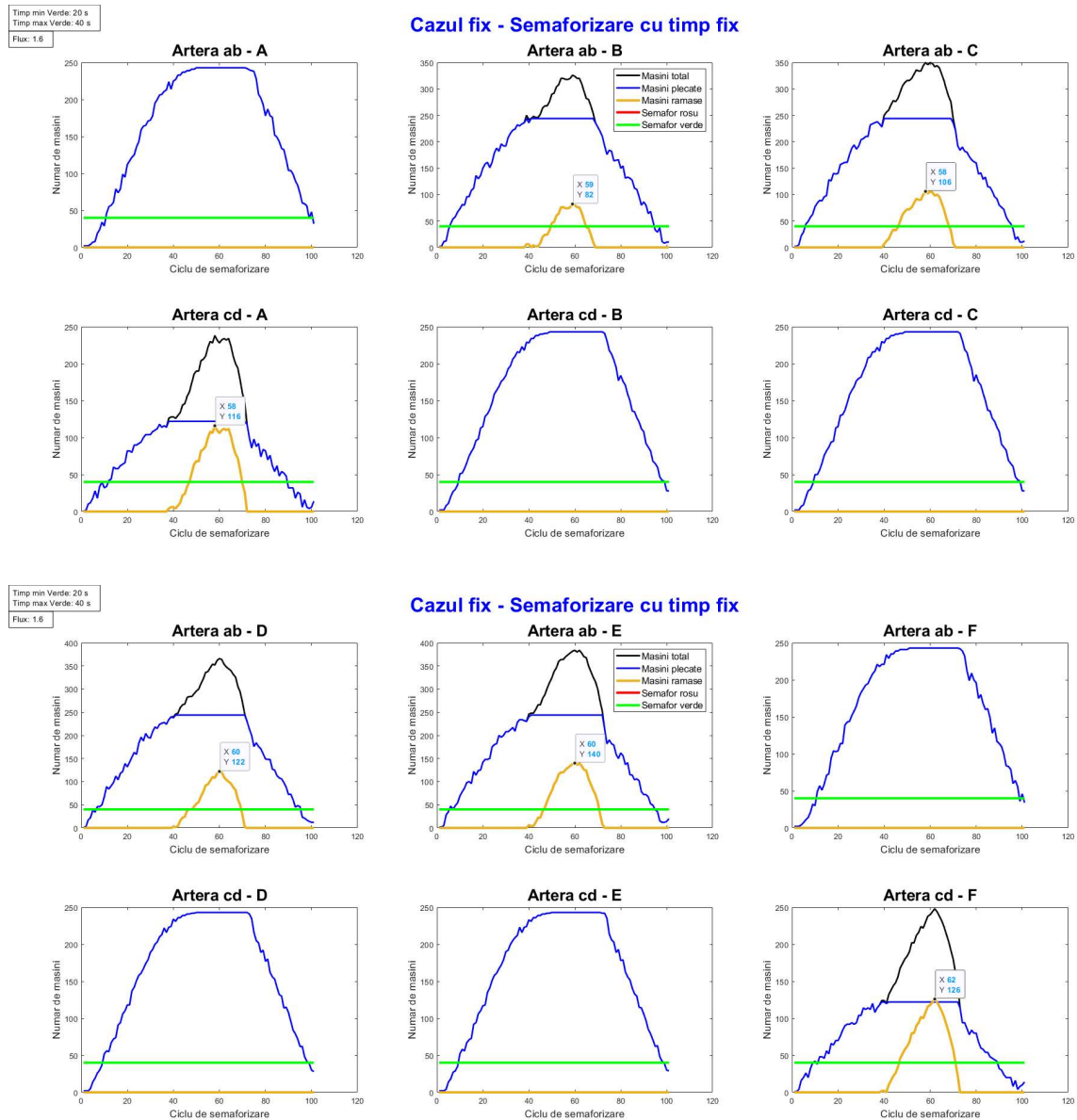


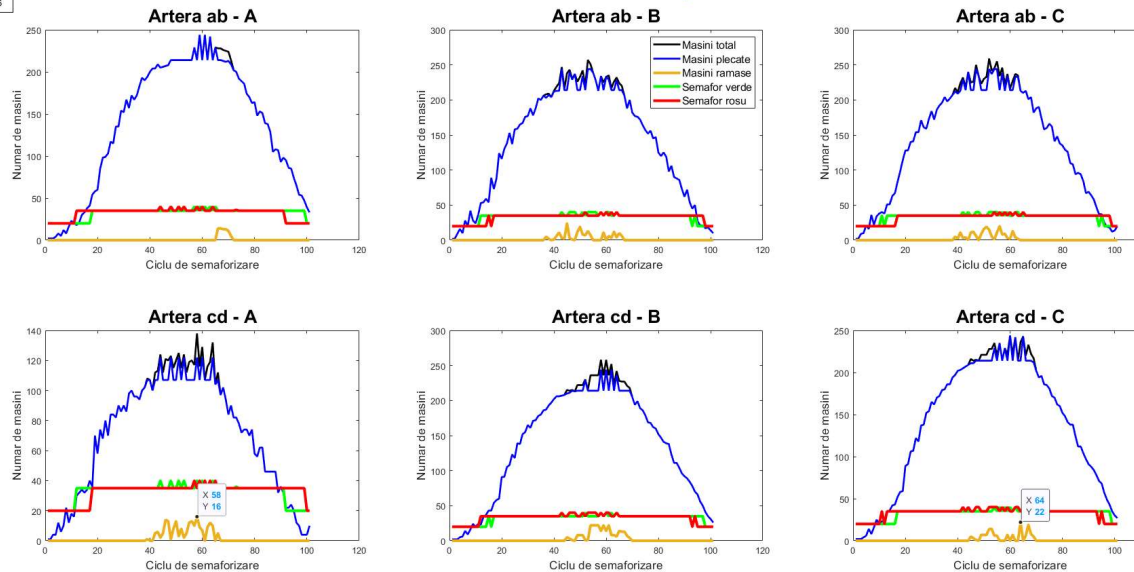
Figura 6. Evoluția numărului de mașini pentru cele 6 intersecții pe fiecare arteră.

3.4 Rezultate Q-Learning în comparație cu semaforizarea fixă

În cazul folosirii semaforizării prin metoda Q-Learning, timpul de semaforizare se modifică proporțional cu numărul de mașini implicate în trafic. Diferența între numărul total de mașini (negru) și numărul mașinilor care au trecut (albastru) este foarte mic (galben), aproape de 10 ori mai mic ca în cazul fix, după cum se poate observa în Figura 7.

Timp min Verde: 20 s
 Timp max Verde: 40 s
 Flux: 1.6

Cazul variabil - Semaforizare cu timp variabil folosind QL



Timp min Verde: 20 s
 Timp max Verde: 40 s
 Flux: 1.6

Cazul variabil - Semaforizare cu timp variabil folosind QL

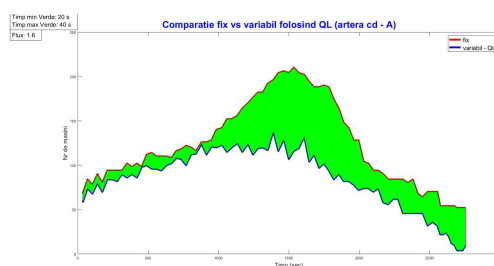
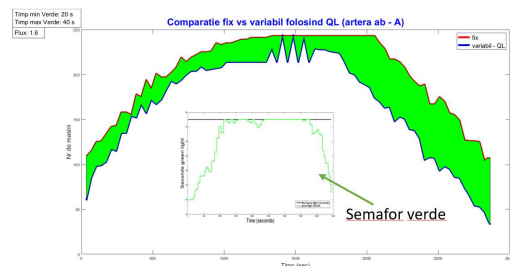
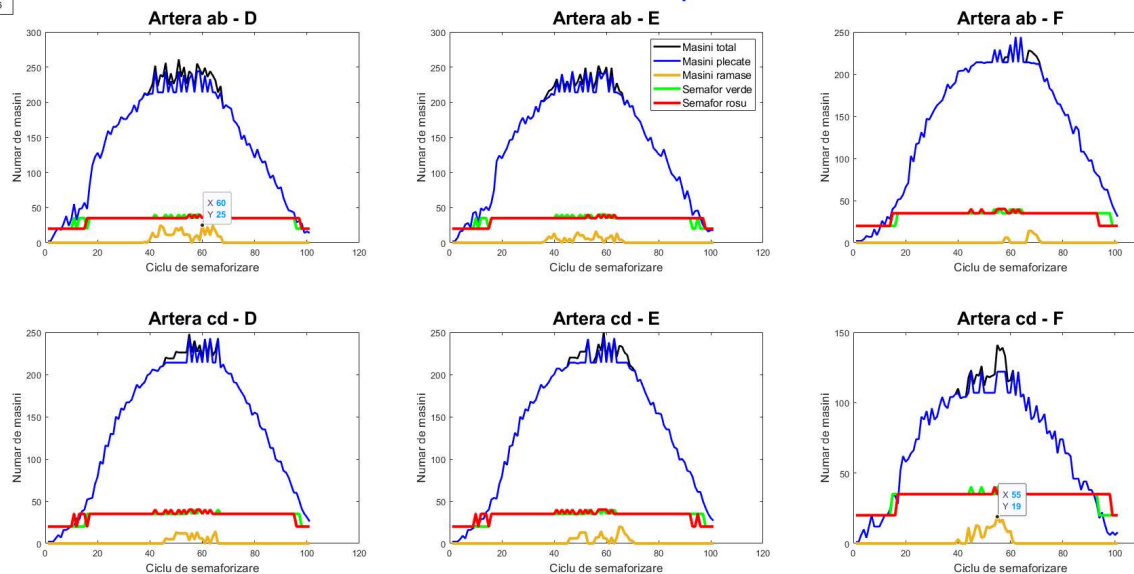


Figura 7. Evoluția numărului de mașini pe fiecare arteră pentru 6 intersecții interconectate utilizând algoritmul QL.

3.5 Rezultate Deep-Learning în comparație cu semaforizarea fixă

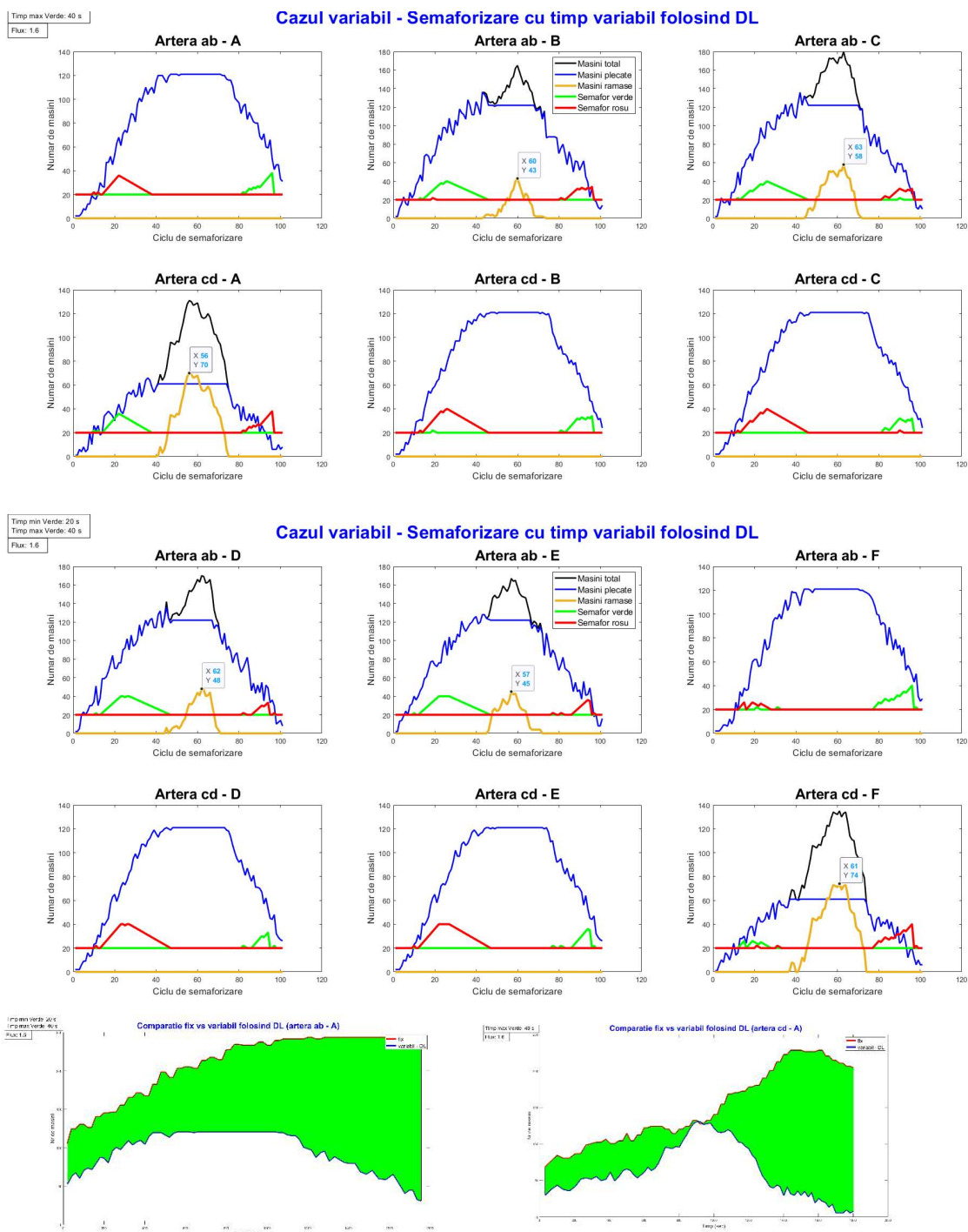


Figura 8. Evoluția numărului de mașini pe fiecare arteră pentru 6 intersecții interconectate utilizând algoritmul DL.

În cazul Deep Learning rețeaua antrenată menține timpul minim pe perioada traficului de vârf, chiar dacă rămân mașini care nu trec și se formează aglomerație. Aceasta aglomerație este mai mică decât în cazul fix (se poate vedea diferența în Figura 8). La un moment dat se ajunge ca numărul de mașini total implicate în trafic în cazul DL să fie aproape egal cu cel în cazul fix (figura , artera ab), însă avem un avantaj al timpului de așteptare care scade considerabil în cazul DL. Comparativ cu cazul fix cu ajutorul DL avem mai puține mașini rămase (medie 74 - DL vs 117 - fix) dar totuși sunt mai multe decât la QL(31). Acest lucru se datorează menținerii semaforului pe valoarea minimă de verde pe durata traficului de vârf timp în care o parte dintre vehicule nu reușesc să treacă în primul ciclu de semaforizare. Acest lucru va avea efecte benefice asupra timpului de deplasare.

4 Concluzii

Utilizarea tehnicilor de învățare automată (QL și DL) reduce congestia pe ambele artere. Utilizarea acestui tip de semafor are efecte pozitive în oraș dacă majoritatea intersecțiilor mari sunt conectate la un sistem central. Dacă un anumit drum este supraaglomerat atunci semafoarele nu pot optimiza traficul și este necesar să se intervină asupra infrastructurii (ex. introducerea unui sens giratoriu). În Tabelul 1 este reprezentată reducerea congestiei și a timpului de așteptare folosind QL pentru un număr de intersecții interconectate în creștere. Comparția este făcută față de semaforizarea cu timp fix.

Tabel 1. Gradul de reducere a congestiei raportat la numărul de noduri

Numar de intersectii interconectate	Reducerea congestiei	Reducerea timpului de asteptare
1 intersectie	50.18 %	48 s
3 intersectii	36.99 %	41 s
6 intersectii	30.75%	56.5 s

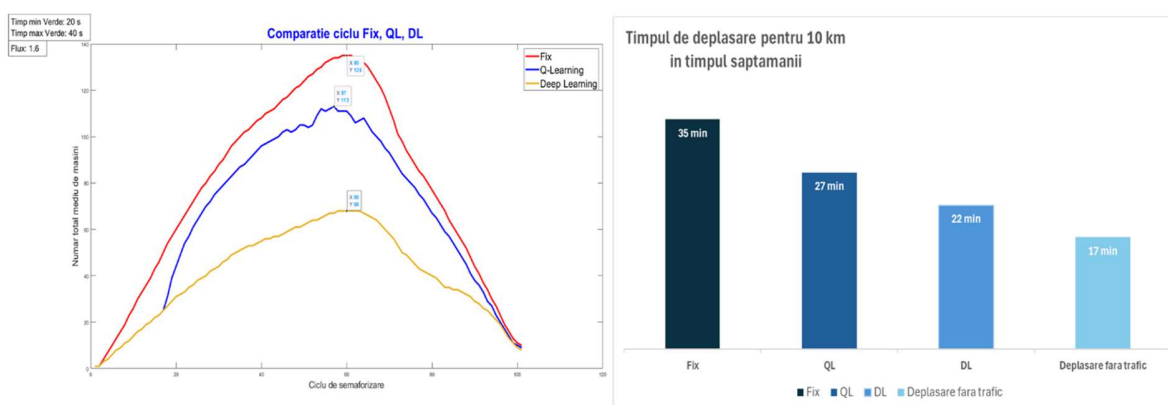


Figura 9. Reducerea timpului de deplasare.

După cum se poate vedea și în Figura 9 folosind algoritmi de inteligență artificială numărul total de mașini la perioada de vârf este redusă în funcție de flux de la sute la zeci de mașini (valorile medii sunt calculate pentru 600 de iterații). Cea mai mare reducere a numărului total de mașini are loc la Deep Learning (graficul galben, datorită alegerii timpului minim de roșu pentru perioada de

vârf, numărul de vehicule acumulându-se proporțional cu timpul de roșu), apoi Q- Learning (graficul albastru) și cea mai mare aglomerație se petrece în cazul fix (graficul roșu) care este metoda folosită astăzi în București și majoritatea orașelor lumii.

Prin utilizarea timpului variabil de semaforizare se reduce și timpul de staționare în trafic și implicit timpul de deplasare. În figura este reprezentat timpul total de deplasare pentru 10 km folosind cele trei metode. Pentru cazul fix timpul de deplasare pentru 10 km durează dublu față de deplasarea fără trafic. QL produce un timp de deplasare mai mic, de 27 de minute și are și un număr de mașini rămase destul de mic comparativ cu cazul fix. DL duce la un timp destul de convenabil de deplasare, de 22 de minute în condiții de trafic, cu mențiunea că la perioada de vârf este posibil să nu treacă toate mașinile în primul ciclu de semaforizare și să apară unele aglomerații, mai mici însă decât în cazul cu timp fixat. Deplasarea fără trafic care corespunde timpului de 17 minute în care un șofer ar parcurge cu viteza legală distanță de 10 km fără a trece prin trafic sau sisteme de semaforizare. Astfel șoferii vor petrece mai puțin timp în trafic și poluarea va fi mai mică.

Bibliografie

- [1] B.S. Kerner, Phase transitions in traffic flow, Traffic and Granular Flow, Springer Verlag, 2000, pp. 253-283.
- [2] Inteligența Artificială, Algoritmul Q-Learning EITCA.org - [link](#)
- [3] Michael I. Jordan and Tom M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. Science 349, 6245 (2015), 255–260.
- [4] Q-Learning Algorithms: A Comprehensive Classification and Applications s, IEEE, septembrie 2019 - [link](#)
- [5] Gerald Tesauro. 1995. Temporal difference learning and TD-Gammon. Communications of the ACM 38, 3 (1995), 58–68.
- [6] Rețele neurale recurente - RNN, LTSM, unit.AI - [link](#)
- [7] ADI (2024). traffic_dataset.mat (https://www.mathworks.com/matlabcentral/fileexchange/154840-traffic_dataset-mat), MATLAB Central File Exchange. Retrieved August 6, 2024.

Dr.ing. Mihaela-Ancuța Mone

Drd.ing. Marius Teme

Director proiect,
S.l.dr.ing. Cătălin Dimon