ns-3

Workshop on ns-3 2023

28 – 29 June 2023 | Arlington, VA, USA

Integration of Machine Learning with ns-3: Challenges and Opportunities

Eduardo Nuno Almeida

INESCTEC

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Portugal

OUTLINE

Machine Learning Background

Integration of Machine Learning with ns-3

Train Machine Learning Models using ns-3

Conclusions

MACHINE LEARNING CATEGORIES

SUPERVISED LEARNING

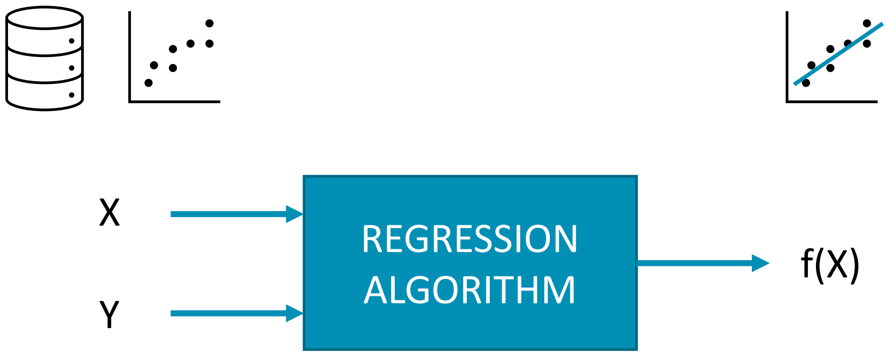
UNSUPERVISED LEARNING

REINFORCEMENT LEARNING

SUPERVISED LEARNING

LEARN FUNCTION MAPPING FEATURES (INPUT X) TO LABELS (OUTPUT Y)

REGRESSION ALGORITHMS



X

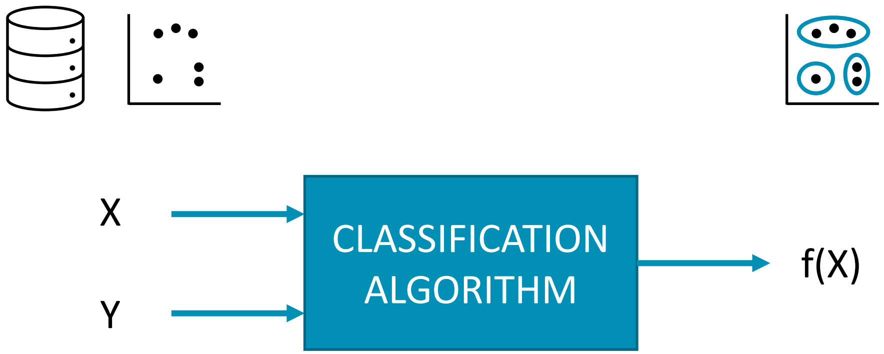
Y

REGRESSION ALGORITHM

f(X)

Estimate Y (Output) for X (Input)

CLASSIFICATION ALGORITHMS



X

Y

CLASSIFICATION ALGORITHM

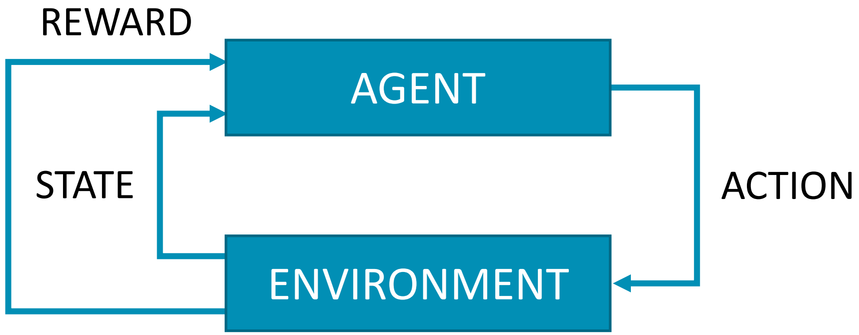
f(X)

Classify Data into Finite Categories

REINFORCEMENT LEARNING

TRAIN AGENT TO LEARN OPTIMAL POLICY TO MAXIMIZE EPISODE’S CUMULATIVE REWARD

- Policy: action to take for a given state
- Maximize episode’s cumulative reward
- Episode: State → Action → Reward



✓ Learn and adapt to scenario dynamics

✓ Real-time network performance metrics

✓ Learn from experience

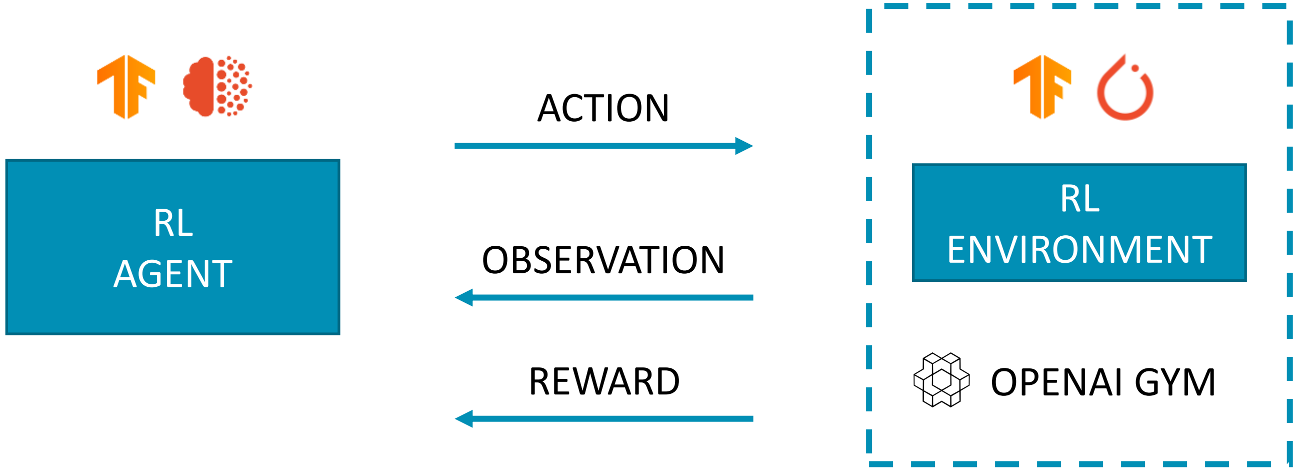
✗ Learning requires many episodes

✗ Requires realistic interactive environment

- Challenging to train agents in testbeds
- ns-3 can serve as environment

OPENAI GYM

- Standard Python API for Reinforcement Learning
- Manage interaction between RL agent and environment
- Independent of agent's implementation
  - Allows fair and easy comparison between RL algorithms



INTEGRATION OF ML WITH NS-3

INTEGRATION WITH ML FRAMEWORKS

- Integration via third-party modules
  - ns3-gym, ns3-ai

✓ Integration with existing ML frameworks in Python

✓ Reuse existing ML models

✗ Computational performance overhead
  - Due to data exchange between processes

NATIVE INTEGRATION

- No native integrations
  - E.g., ONNX framework

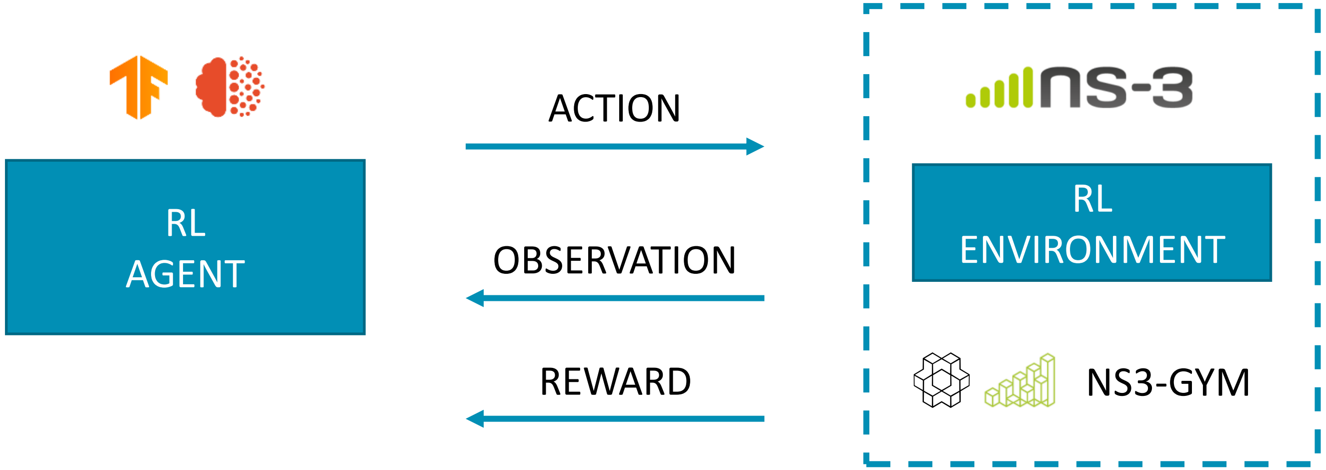
✓ Improved computational performance
  - No overhead due to data exchange

✗ Additional dependency to manage

✗ Tight coupling of ns-3 and ML code
  - Code recompilation for ML model updates
  - ONNX separates runtime from ML model

NS3-GYM MODULE

- Development of OpenAI Gym RL environments in ns-3
  - Execute actions, provide observations and reward in underlying ns-3 simulation
  - Data exchanged via protobuf messages over ZMQ / sockets



P. Gawłowicz and A. Zubow, “ns-3 meets OpenAI Gym: The playground for machine learning in networking research,” in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2019, pp. 1–6.

NS3-GYM MODULE ANALYSIS

ADVANTAGES

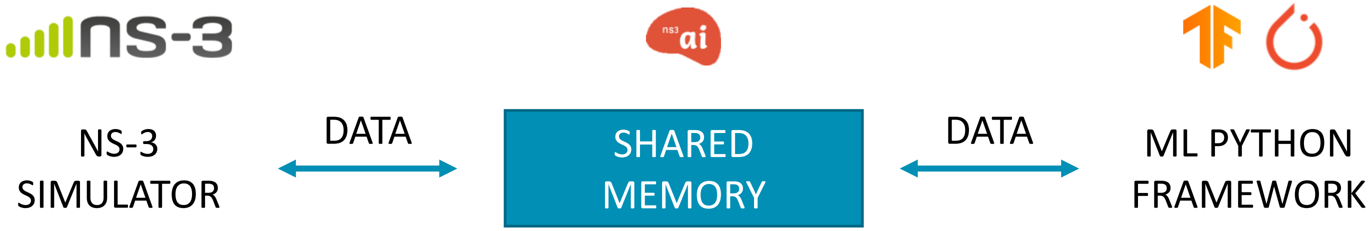
- Seamless integration with OpenAI Gym
- Helper scripts to launch ns-3 and RL agent
- Examples provided by the module
- Community on GitHub and ns-3-users

CHALLENGES

- Multiple issues reported in GitHub
  - No support for matrix values
  - No reshaping of Box container
  - No check if values are within defined range
- Rare updates to module
- Documentation only available in the paper
  - No quick-start guides or tutorials
- Computational overhead due to sockets

NS3-AI MODULE

- Integration with existing Python ML frameworks
  - API to read and write data between ns-3 and ML process
  - Data exchange via shared memory



NS-3 SIMULATOR ⟷ DATA ⟷ SHARED MEMORY ⟷ DATA ⟷ ML PYTHON FRAMEWORK

H. Yin et al., *“NS3-AI: Fostering artificial intelligence algorithms for networking research,”* in Proceedings of the 2020 Workshop on ns-3, 2020, pp. 57–64.

NS3-AI MODULE ANALYSIS

ADVANTAGES

- Easy integration with ML frameworks and ns-3
- Flexible and powerful data exchange mechanism
  - Can be extended beyond AI applications
- Ongoing GSoC 2023 to improve ns3-ai
  - OpenAI Gym interface, performance, …
- Good documentation and examples
- Community on GitHub and ns-3-users

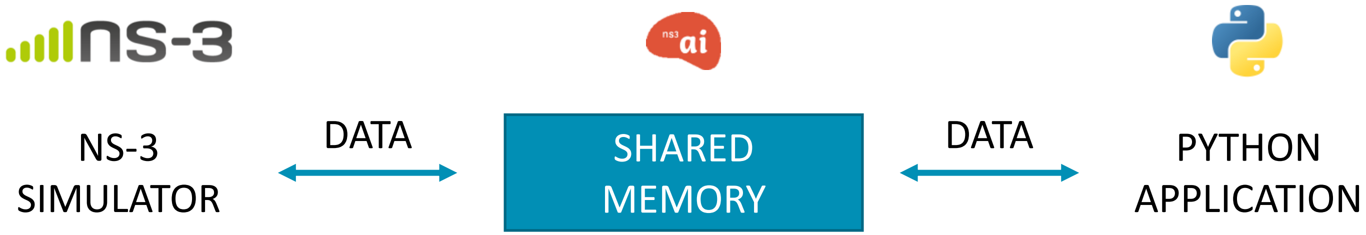
CHALLENGES

- Ongoing fixes / improvements to main issues
- No integration with applications other than Python
- No helper scripts to launch ns-3 and ML application

GSoC 23 ns3-ai. https://www.nsnam.org/wiki/GSOC2023ns3-ai

NS3-AI ADDITIONAL USE CASES

- Shared memory mechanism can be used in scenarios beyond AI
- Enables integration with any external Python application
  - Optimization solvers
  - Real applications (e.g., network controllers)



NS-3 SIMULATOR

DATA

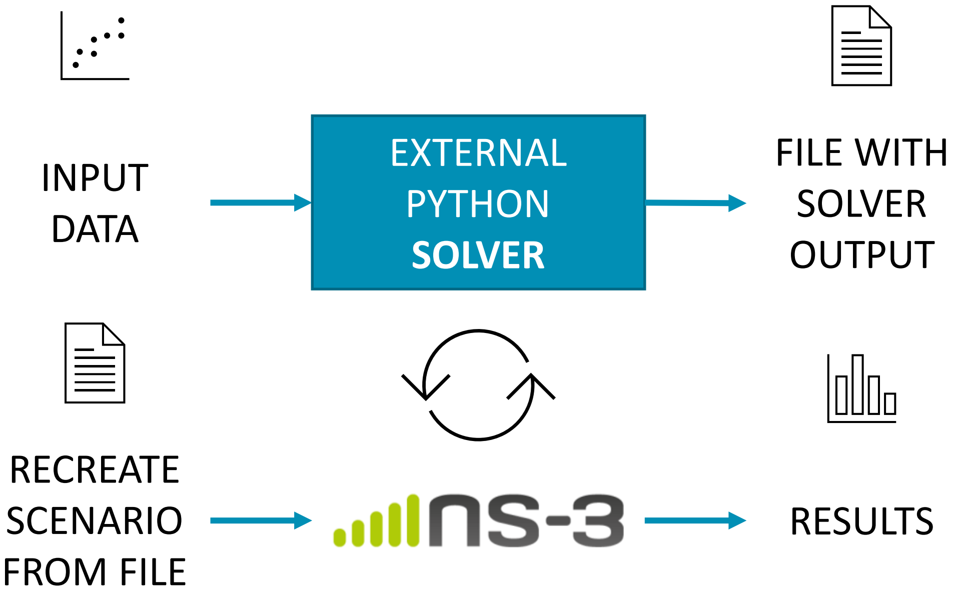
SHARED MEMORY

DATA

PYTHON APPLICATION

INTEGRATION OF SOLVERS WITH NS3-AI

OFFLINE METHODOLOGY



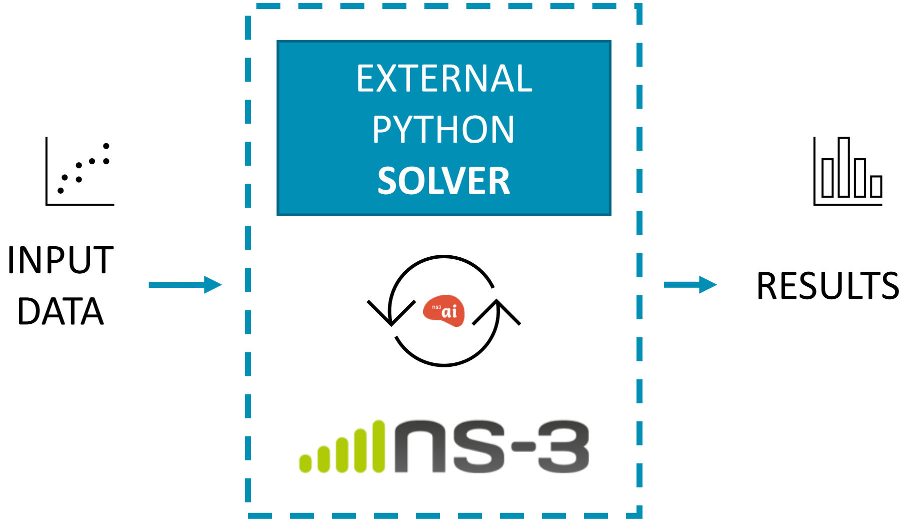
INPUT DATA → EXTERNAL PYTHON **SOLVER** → FILE WITH SOLVER OUTPUT

RECREATE SCENARIO FROM FILE → ns-3 → RESULTS

No Interaction between Solver and ns-3.

Create Offline Simulation Replicating Solver Output.

ONLINE METHODOLOGY



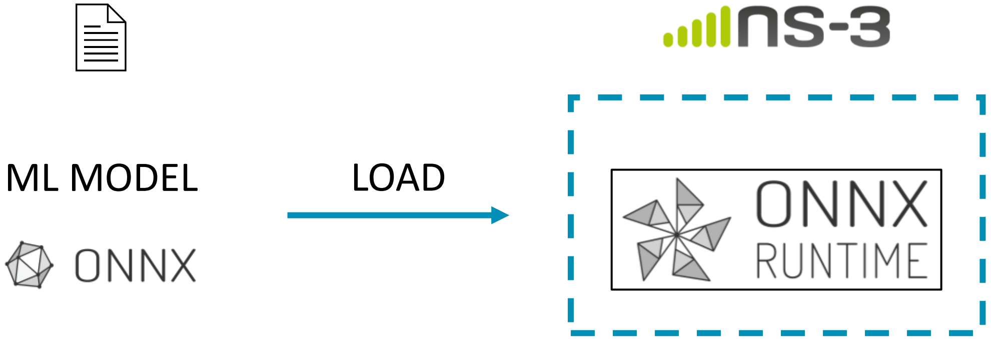
INPUT DATA → EXTERNAL PYTHON **SOLVER** / ns-3 → RESULTS

Real-time Interaction between Solver and ns-3.

Dynamic Simulations Based on Solver Output.

ONNX FRAMEWORK

- Open Neural Network Exchange (ONNX)
  - Open format to represent ML models
  - Portable and interoperable among platforms and frameworks
- Use cases
  - Build and share of ML models
  - Deployment of ML models for inference using ONNX runtime



TRAINING ML MODELS USING NS-3

GENERATE DATASETS FOR SUPERVISED LEARNING

- When experimental data not available
- When insuficient experimental data
- Augment / transform existing experimental dataset
  - Collect results for different scenario parameters

CREATE REALISTIC RL ENVIRONMENTS

- Realistic interactive environment for RL
- Train RL agents with offline learning
- Pre-train / improve policies for online learning
- Evaluate and compare RL trained policies

TRAINING ML MODELS USING NS-3

CHALLENGES

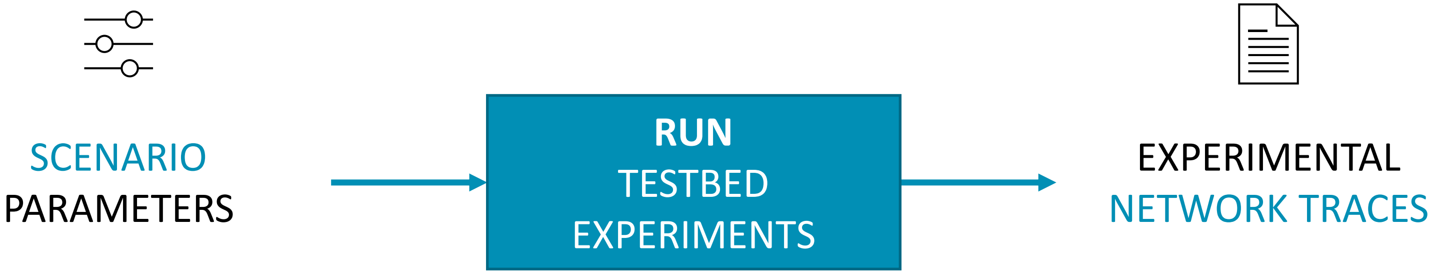
- Existing models in ns-3 may not fully capture environment dynamics
  - Extreme scenarios
- Non-existent models
- Generate realistic datasets with randomness and noise
- Computational performance

OPPORTUNITIES

- Improve ns-3 models with trace-based or ML
  - Collect experimental data in testbed
  - Accurate and customized models
  - Specific to scenario
- Trace-based simulation approaches
  - Accurate, repeatable and reproducible
  - Propagation loss, channel occupancy, rate adaptation, MIMO, …
- ML-based models
  - ML Propagation Loss (MLPL) model

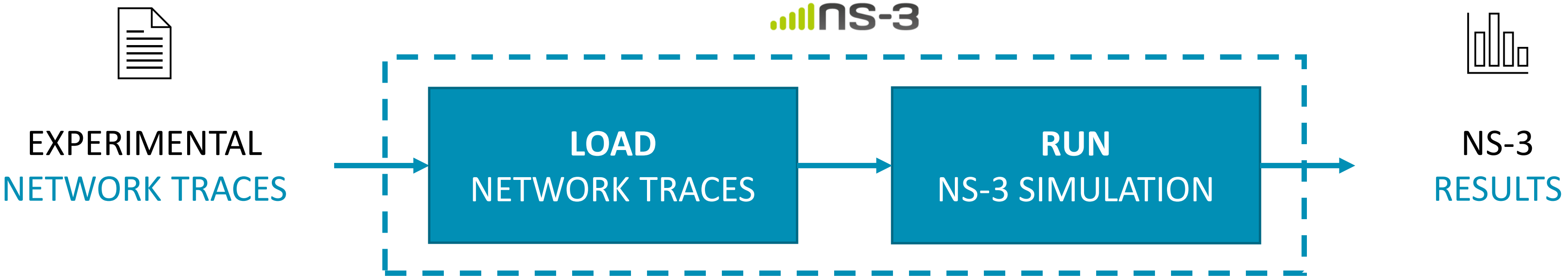
TRACE-BASED SIMULATION APPROACH

NETWORK TRACES COLLECTION



REPEAT AND REPRODUCE EXACT EXPERIMENTAL CONDITIONS IN NS-3

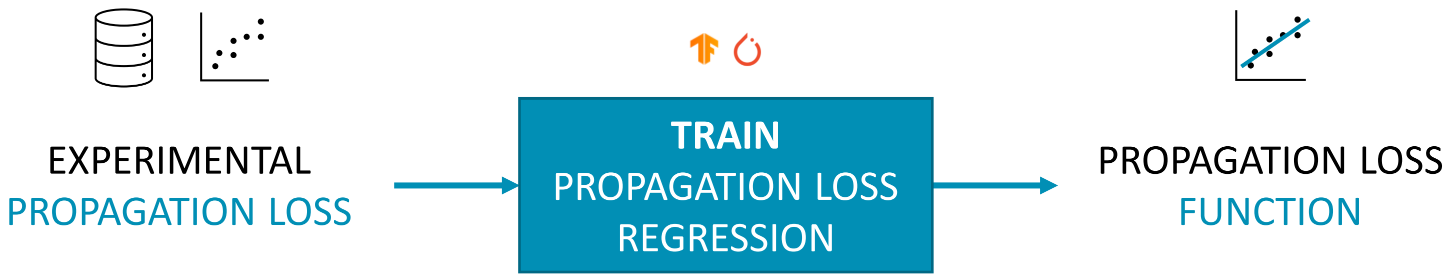
NETWORK TRACES USAGE



H. Fontes, R. Campos, and M. Ricardo, "A Trace-based ns-3 Simulation Approach for Perpetuating Real-World Experiments", *in Proceedings of the 2017 Workshop on ns-3 (WNS3 '17)*, pp. 118–124

ML PROPAGATION LOSS (MLPL) MODEL

TRAINING



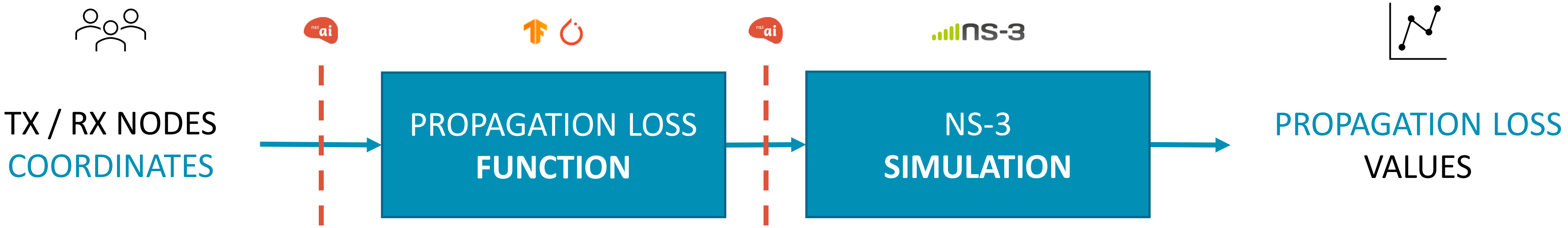
EXPERIMENTAL PROPAGATION LOSS

TRAIN PROPAGATION LOSS REGRESSION

PROPAGATION LOSS FUNCTION

ML MODEL TRAINED WITH EXPERIMENTAL NETWORK TRACES

ESTIMATION



TX / RX NODES COORDINATES

PROPAGATION LOSS FUNCTION

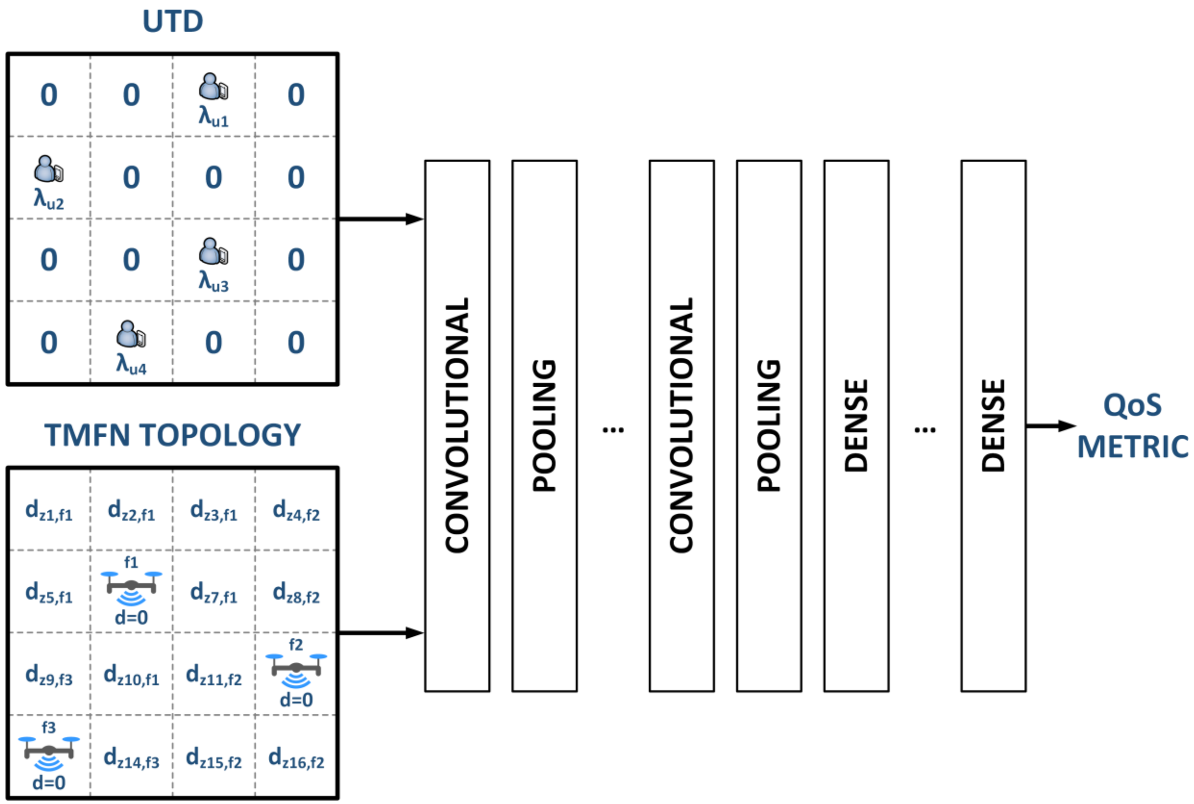
NS-3 SIMULATION

PROPAGATION LOSS VALUES

E. N. Almeida, et al., “Position-Based Machine Learning Propagation Loss Model Enabling Fast Digital Twins of Wireless Networks in ns-3”, *in Proceedings of the 2023 Workshop on ns-3 (WNS3 ‘23)*, pp. 69–77

TRAINING ML MODELS USING NS-3

ML QUALITY OF SERVICE ESTIMATOR

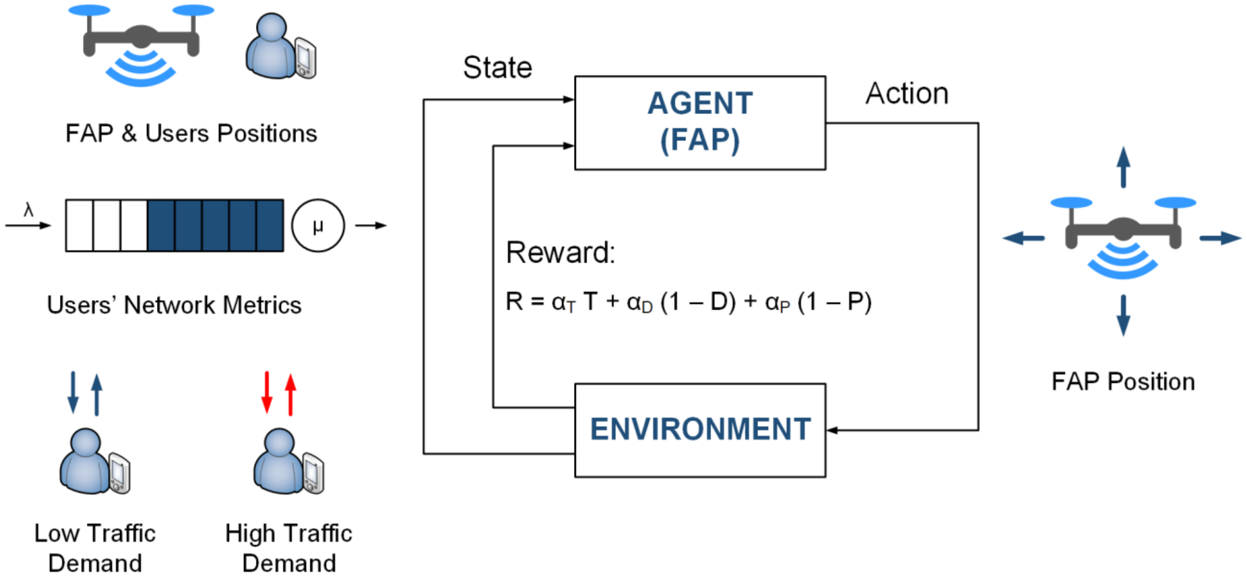

UTD
0 0 λ_{u1} 0
λ_{u2} 0 0 0
0 0 λ_{u3} 0
0 λ_{u4} 0 0
TMFN TOPOLOGY
$d_{z1,f1}$ $d_{z2,f1}$ $d_{z3,f1}$ $d_{z4,f2}$
$d_{z5,f1}$ f1 d=0 $d_{z7,f1}$ $d_{z8,f2}$
$d_{z9,f3}$ $d_{z10,f1}$ $d_{z11,f2}$ f2 d=0
f3 d=0 $d_{z14,f3}$ $d_{z15,f2}$ $d_{z16,f2}$
CONVOLUTIONAL
POOLING
...
CONVOLUTIONAL
POOLING
DENSE
...
DENSE
QoS METRIC

- Estimate QoS based on
  - Users traffic demand
  - UAV positions
- Convolutional neural network
- One estimator per QoS metric
  - Throughput, Delay, PLR
- Dataset generated in ns-3

E. N. Almeida et al., "A Machine Learning Based Quality of Service Estimator for Aerial Wireless Networks," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 1-6

TRAINING ML MODELS USING NS-3

DRL TRAFFIC-AWARE UAV PLACEMENT

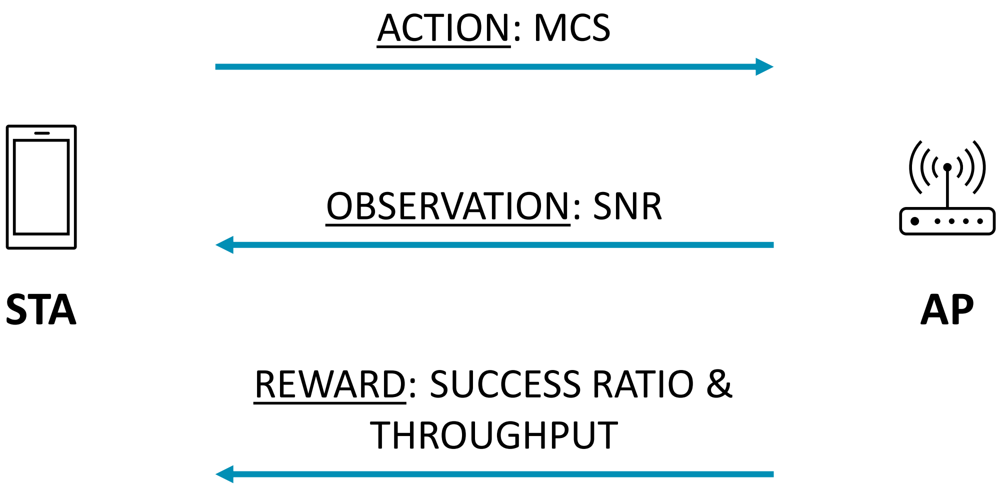


- Position FAP
- According to users traffic demand
- Maximize network utility
- Trained and evaluated with ns3-gym

E. N. Almeida, R. Campos, and M. Ricardo, "Traffic-Aware UAV Placement using a Generalizable Deep Reinforcement Learning Methodology," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–6

TRAINING ML MODELS USING NS-3

DRL DATA-DRIVEN WI-FI RATE ADAPTATION



ACTION: MCS

OBSERVATION: SNR

REWARD: SUCCESS RATIO & THROUGHPUT

$$R = \frac{\mathrm{MCS}_n}{\mathrm{MCS}_7} \times \mathrm{FSR}, \qquad n \in \{1, \dots, 7\}$$

- Modulation and Coding Scheme (MCS)
- According to channel state
- Maximize throughput and Frame Success Ratio (FSR)

- Trained and evaluated with ns3-gym + trace-based

R. Queirós, E. N. Almeida, H. Fontes, J. Ruela, and R. Campos, “Wi-Fi Rate Adaptation using a Simple Deep Reinforcement Learning Approach,” in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–3

CONCLUSIONS

- Integration of external ML frameworks via ns3-ai and ns3-gym
  - Opportunity to improve the modules
  - Consider supporting ONNX for deployment of ML models

- ns3-ai powerful tool for applications beyond AI
  - Integration with Python applications (e.g., solvers or controllers)

- ns-3 interesting tool to train and evaluate ML models
  - Generate training datasets for supervised learning
  - Create realistic interactive environments for RL
  - Can be enhanced with trace-based or ML-based models

QUESTIONS?

Integration of Machine Learning with ns-3: Challenges and Opportunities

Eduardo Nuno Almeida | INESC TEC & FEUP, Portugal
eduardo.n.almeida@inesctec.pt