Machine Learning and Wi-Fi: Confluences, Ongoing Activities, and Ways Forward

https://mlwifitutorial.github.io/

Francesc Wilhelmi, Szymon Szott, Katarzyna Kosek-Szott, Boris Bellalta

October 6th, 2023
ACM Mobicom

Francesc Wilhelmi

- Research Engineer at Radio Systems Research (RSR), Nokia Bell-Labs Germany
- PhD from Universitat Pompeu Fabra (UPF)
- Interests:
    - Wi-Fi networks & unlicensed wireless technologies
    - AIML for communications
    - DLT & blockchain
- Contact: francisco.wilhelmi@nokia.com

NOKIA BELL LABS

Szymon Szott

- Associate professor at AGH University of Kraków, Poland
- Research interests
  - Wireless local area networks
  - Channel access
  - Quality of service
  - Security
  - Inter-technology coexistence
- **ML interest:** clever ways to improve performance
- Contact:
  - szott@agh.edu.pl
  - https://szymonszott.github.io/

Katarzyna Kosek-Szott

- Associate Professor at AGH University of Krakow, Poland
- Main research interests
  - Novel IEEE 802.11 amendments
  - Coexistence of radio technologies in unlicensed bands
  - Application of ML in IEEE 802.11 networks
- Contact: kks@agh.edu.pl

AGH

Boris Bellalta

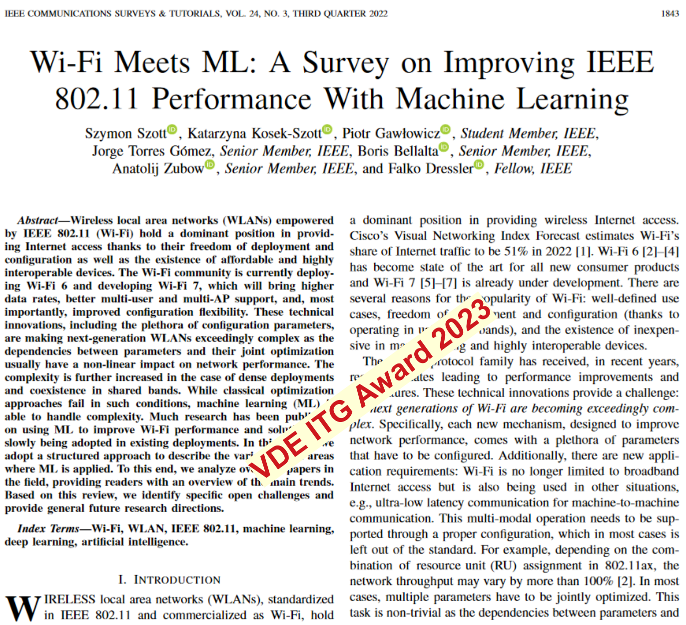
- Full-professor at UPF
- Head of the Wireless Networking group
    - Unlicensed bands, Wi-Fi
    - XR Immersive Communications (over Wi-Fi)
- **ML interest**: ‘intelligent’ adaptive systems; low-complexity but responsive RL agents; interplay between ML agents;
- Contact: boris.bellalta@upf.edu

Universitat
Pompeu Fabra
Barcelona

Acknowledgements

- Piotr Gawłowicz, Jorge Torres Gómez, Anatolij Zubow, Falko Dressler, co-authors of the survey paper "**Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning**" in IEEE Communications Surveys & Tutorials, vol. 24, no. 3, pp. 1843-1893, third quarter 2022, doi: 10.1109/COMST.2022.3179242. [1]

- From projects:
  - National Science Centre, Poland, under Grant DEC-2020/39/I/ST7/01457
  - WI-XR PID2021-123995NB-I00 (MCIU/AEI/FEDER,UE)



IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 24, NO. 3, THIRD QUARTER 2022 1843

Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance With Machine Learning

Szymon Szott, Katarzyna Kosek-Szott, Piotr Gawłowicz, *Student Member, IEEE*, Jorge Torres Gómez, *Senior Member, IEEE*, Boris Bellalta, *Senior Member, IEEE*, Anatolij Zubow, *Senior Member, IEEE*, and Falko Dressler, *Fellow, IEEE*

***Abstract*—Wireless local area networks (WLANs) empowered by IEEE 802.11 (Wi-Fi) hold a dominant position in providing Internet access thanks to their freedom of deployment and configuration as well as the existence of affordable and highly interoperable devices. The Wi-Fi community is currently deploying Wi-Fi 6 and developing Wi-Fi 7, which will bring higher data rates, better multi-user and multi-AP support, and, most importantly, improved configuration flexibility. These technical innovations, including the plethora of configuration parameters, are making next-generation WLANs exceedingly complex as the dependencies between parameters and their joint optimization usually have a non-linear impact on network performance. The complexity is further increased in the case of dense deployments and coexistence in shared bands. While classical optimization approaches fail in such conditions, machine learning (ML) [illegible] able to handle complexity. Much research has been publ[illegible] on using ML to improve Wi-Fi performance and solu[illegible] slowly being adopted in existing deployments. In thi[illegible] we adopt a structured approach to describe the vari[illegible] areas where ML is applied. To this end, we analyze ov[illegible] papers in the field, providing readers with an overview of th[illegible] main trends. Based on this review, we identify specific open challenges and provide general future research directions.**

***Index Terms*—Wi-Fi, WLAN, IEEE 802.11, machine learning, deep learning, artificial intelligence.**

I. Introduction

WIRELESS local area networks (WLANs), standardized in IEEE 802.11 and commercialized as Wi-Fi, hold a dominant position in providing wireless Internet access. Cisco's Visual Networking Index Forecast estimates Wi-Fi's share of Internet traffic to be 51% in 2022 [1]. Wi-Fi 6 [2]–[4] has become state of the art for all new consumer products and Wi-Fi 7 [5]–[7] is already under development. There are several reasons for th[illegible]opularity of Wi-Fi: well-defined use cases, freedom of [illegible]ment and configuration (thanks to operating in u[illegible] bands), and the existence of inexpensive in ma[illegible]g and highly interoperable devices.

The [illegible] protocol family has received, in recent years, re[illegible]ates leading to performance improvements and [illegible]ures. These technical innovations provide a challenge: [illegible] *next generations of Wi-Fi are becoming exceedingly complex*. Specifically, each new mechanism, designed to improve network performance, comes with a plethora of parameters that have to be configured. Additionally, there are new application requirements: Wi-Fi is no longer limited to broadband Internet access but is also being used in other situations, e.g., ultra-low latency communication for machine-to-machine communication. This multi-modal operation needs to be supported through a proper configuration, which in most cases is left out of the standard. For example, depending on the combination of resource unit (RU) assignment in 802.11ax, the network throughput may vary by more than 100% [2]. In most cases, multiple parameters have to be jointly optimized. This task is non-trivial as the dependencies between parameters and

VDE ITG Award 2023

NATIONAL SCIENCE CENTRE POLAND

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Unión Europea
Fondo Europeo de Desarrollo Regional
"Una manera de hacer Europa"
AGENCIA ESTATAL DE INVESTIGACIÓN

Outline

<u>Part 1</u>: **Introduction. Why Wi-Fi may want to adopt AI/ML?** (30 mins) - Katarzyna

- Wi-Fi overview: A 30-year path from IEEE 802.11b to IEEE 802.11be, and beyond.
- Open challenges in Wi-Fi: Dealing with complexity and uncertainty.
- Requirements of next-gen WiFi networks.

<u>Part 2</u>: **A primer on AI/ML** (45 min) - Szymon & Boris

- Concepts, definitions, and overview of the main ML types (supervised, unsupervised, and reinforcement learning), including Wi-Fi examples.
- Popular ML techniques and paradigms: deep learning, reinforcement learning, online learning, federated learning.
- Deployment options: architecture, data handling, marketplaces.

Break (10:00-10:30)

<u>Part 3</u>: **Multi-Armed Bandits for Responsive Wi-Fi networks** (45 mins) - Boris & Szymon

- Multi-Armed Bandits: exploitation-exploration trade-off; e-greedy, Thompson Sampling, UCB.
- Examples: Channel Selection; AP selection;
- Reinforced-lib + Example (MCS selection).

<u>Part 4</u>: **Wi-Fi & ML: Practical considerations** (45 mins) - Francesc

- Wi-Fi & ML: A two-sided relationship
- Adoption of ML in Wi-Fi
- Hands-on Exercise II: Predicting the performance of Wi-Fi through Deep Learning

<u>Part 5</u>: **Open challenges, future research directions, and summary** (15 mins) - Katarzyna

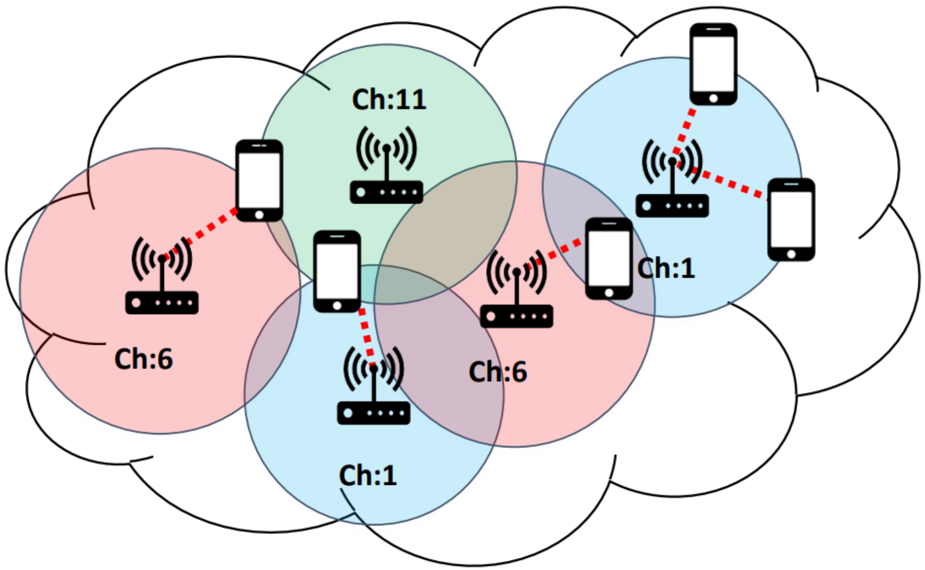
- Open challenges and future research directions; Synergies with other disruptive technologies.
- Current AI/ML trends: LLM and Generative AI.
- Summary and takeaways.

1 Introduction

Why Wi-Fi may want to adopt AI/ML?

IEEE 802.11 Standard

- **PHY/MAC specs**
  - Data plane
  - Management and control
- Operation in **unlicensed bands**
  - **Decentralized spectrum sharing**
  - Coexistence with other technologies
- **Variety of scenarios**
  - Dense, dynamic, diverse, and uncontrolled
  - Wireless broadband access
  - Real-time communication



Uncertainty
(varying channel conditions, channel access contention)

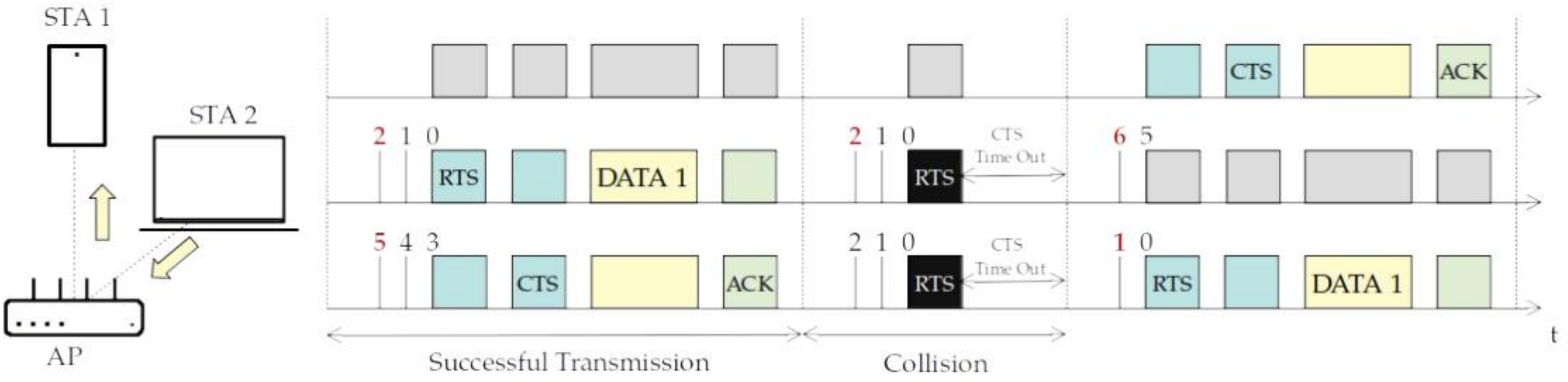
How does Wi-Fi work?

- DCF: CSMA + Backoff (CA) + ARQ
- EDCA: DCF + Multiple Access Categories for Traffic Differentiation

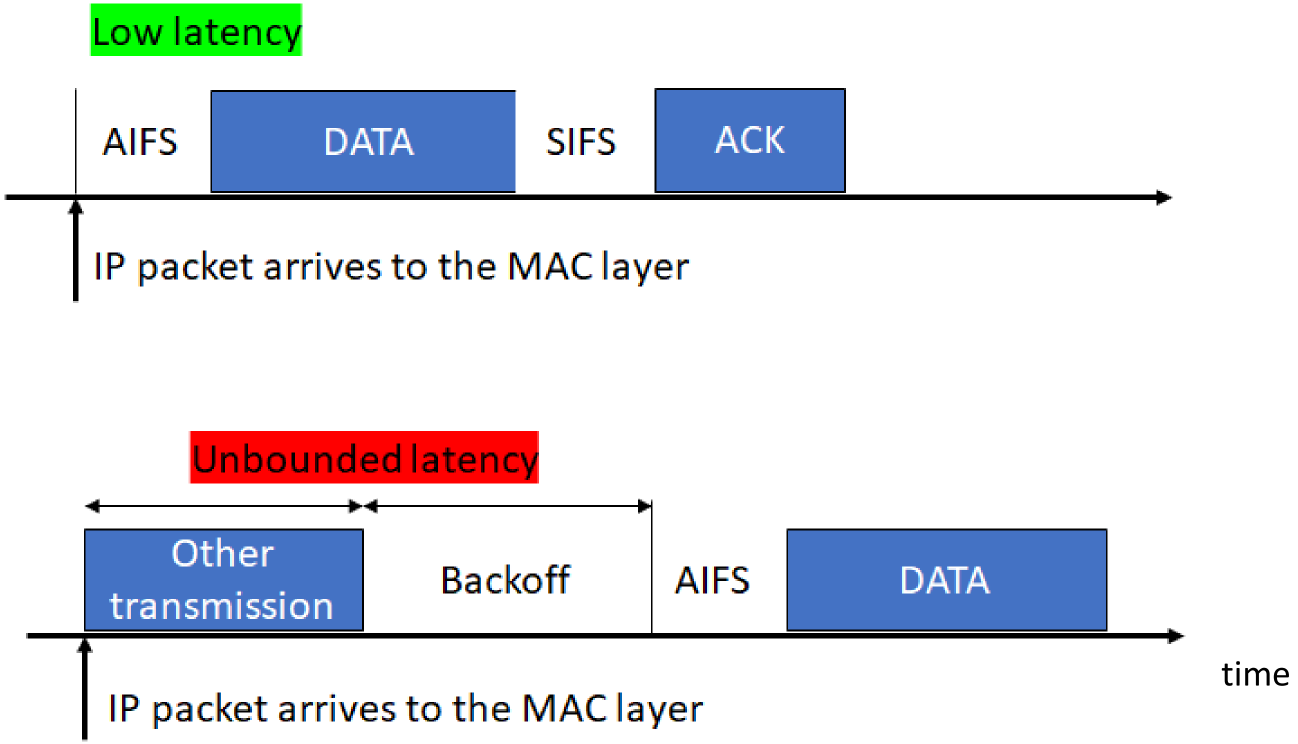
DCF + ARQ

EDCA adds more tx queues

RTS/CTS mechanism reduces collision time

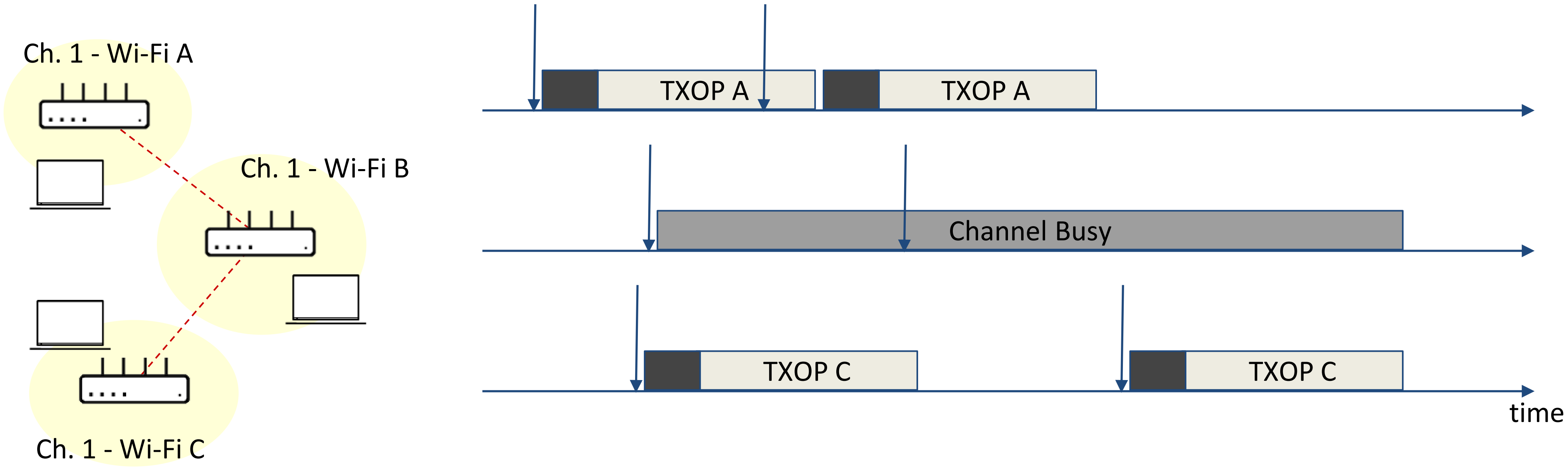


Contention-based Channel Access: Unbounded latency



Coexistence with other unlicensed technologies and Wi-Fi networks

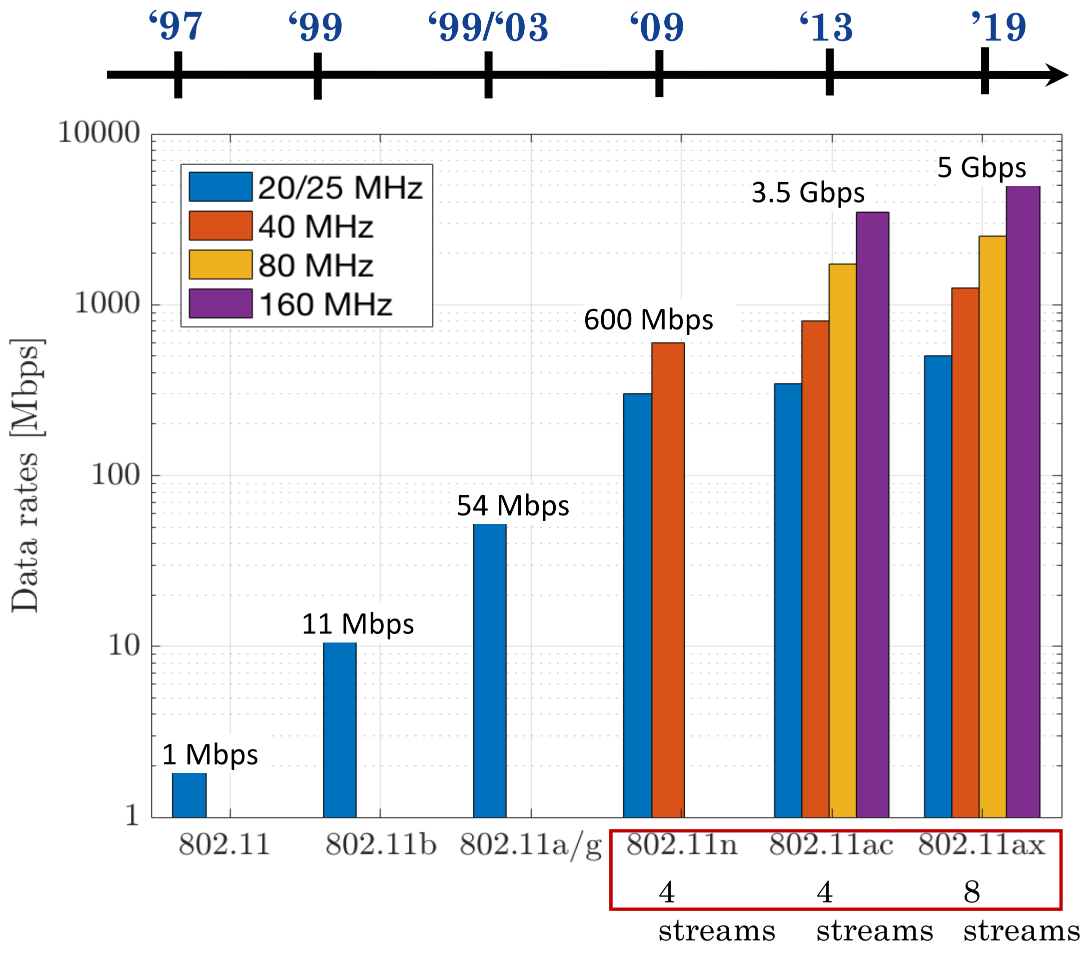
- Performance of a target Wi-Fi network depends on the activity of other Wi-Fi networks within its coverage, and operating in the same channel



Wi-Fi B may not have access to the channel (starvation): low throughput and high delays!

Evolution of Wi-Fi

- **802.11n (Wi-Fi 4) [2.4 & 5 GHz]**
  - Single-user MIMO
  - Packet aggregation
- **802.11ac (Wi-Fi 5) [5 GHz]**
  - Multi-user MIMO (DL)
  - Channel bonding
- **802.11ax (Wi-Fi 6) [5 & 6 GHz]**
  - OFDMA
  - Multi-user MIMO (UL)
- **802.11be (Wi-Fi 7) [2.4, 5 & 6 GHz]**
  - First products expected late 2023
  - <u>Multi-link operation</u>



New 802.11 Amendments

Increased Complexity

- Plethora of parameters
- New mechanisms (e.g., MU transmissions, MLO, MAPC)
- Optimal configuration outside of the standard scope
- Joint parameter optimization: non-linear!

Band [GHz]	2.4, 5, 6, …
Channel number	32, 36, 40, …
Channel width [MHz]	20, 40, 80, …
Guard interval [ns]	400, 800, 1600, …
Coding rate	1/2, 2/3, 3/4, 5/6, …
Modulation	BPSK, QPSK, 16-QAM, 64-QAM, …
Spatial streams	1, 2, …
Contention window	15, 31, 63, …
…	…

Requirements of emerging apps and use cases

Uncertainty of wireless channel

Wi-Fi Use-cases Have Different Requirements

- Broadband Internet access
- Teleconference/video streaming, remote work and education
- Cloud-supported productivity: fast content/file exchange in collaborative environments
- (Cloud) Gaming, VR/AR/Metaverse XR
- Industrial applications: system & plant control (machines, robots, IoT)

Requirements	Low-latency, high-throughput, high-reliability Stable connectivity & predictable performance

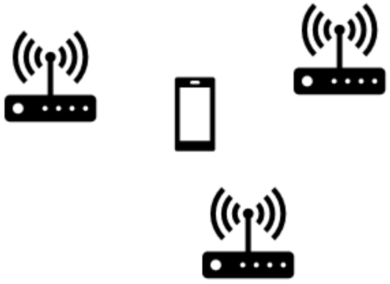
When AI/ML May Be Useful?

- **Adaptation to different environments**
  - Find satisfactory configurations of IEEE 802.11 MAC/PHY parameters
  - Recognize conditions: LOS/NLOS, interferers, coexisting technologies, …
- **Smart decisions**
  - AP selection, wait/transmit (e.g. in MLO), CW/MCS selection, …
- **Improve parameter estimation**
  - Beamforming coefficients, overhead reduction, …
- …

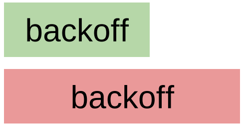
In ML, *“algorithms can learn from training data without being explicitly programmed”*

Performance Trade-offs

- AP selection (if multiple APs available, on different channels)
  - RSSI only: Better channel quality, **risk to overload** a single AP;
  - Load-based: **Low channel quality**;

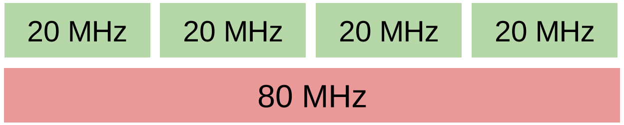


- Channel access parameters (CWmin, CWmax)
  - Low values: fast channel access with low number of contenders (low delay and high throughput), the opposite otherwise; **may be problematic in case of hidden nodes**;



- Spatial Reuse: Increase PD threshold & Decrease Tx. Power
  - Higher chances to transmit (low delay & high throughput);
  - **Lower MCSs** (longer transmissions, lower throughput);

- Using wide channels:
  - Higher rates (high throughput, low delay), but **higher contention** (the opposite);

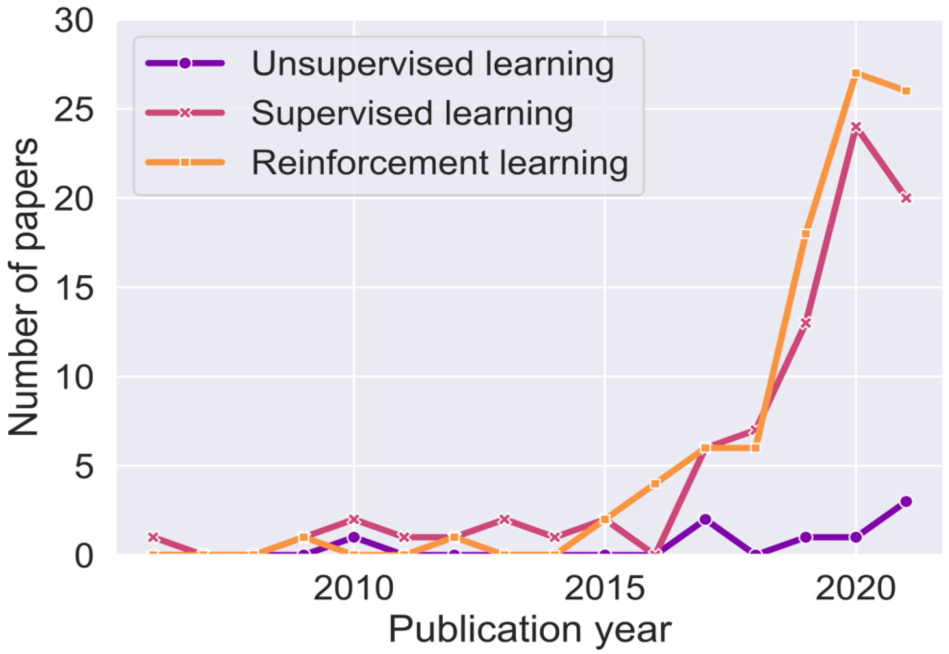


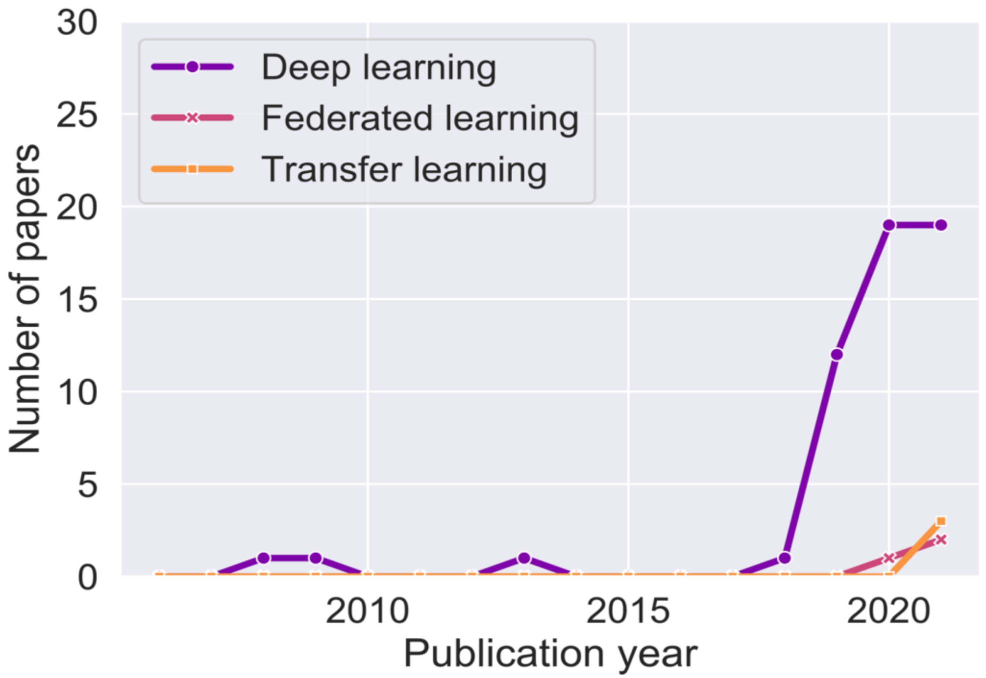
ML Applications to 802.11 Performance Improvement

802.11 area	ML objective	ML methods	Improvement
Channel access	Select CW value, modify CW update rule	Mostly RL	Higher throughput
Link adaptation	Select transmission rate Predict link-layer throughput	Mostly RL SL	Higher throughput
PHY	Classify signal source, de-noise signals estimate interference	Mostly SL	Higher accuracy
Beamforming	Select beam sector, beamwidth, transmit power, predict channel characteristics	Mostly SL	Higher throughput, reduced exploration time
Multi-user operation	Select users, schedule RUs	SL and RL	Higher throughput
Spatial reuse, channel bonding	Select channel, transmit power, carrier sense threshold	Mostly RL	Higher throughput, lower latency

More details in our survey: https://ieeexplore.ieee.org/document/9786784

Which ML methods are used?



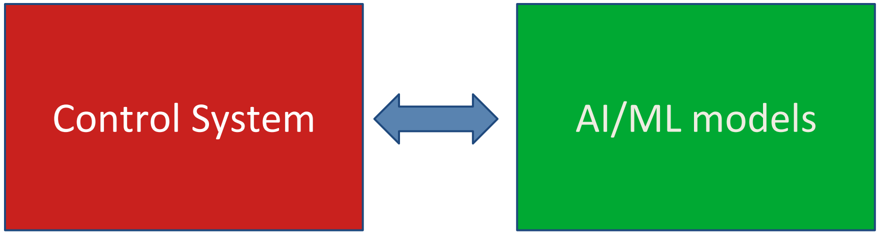


Wi-Fi augmented with AI/ML

- Should we expect that the use of ML techniques will…
    - **improve raw ‘performance’** (i.e., throughput, delay and reliability)?
    - **improve QoE** in general (i.e., better AP selection)?
    - contribute to **avoid BSS ‘starvation’** in dense scenarios?
    - contribute to **reduce ‘overheads’** by reducing the amount of exchanged data?
    - contribute to **improve the performance** of some functionalities?
- Should we achieve any **quantitative targets/explicit gains**?

Control System + AI/ML models

- To use AI/ML models we often have to also develop a control system (code) able to get the most of them.
- This control system uses AI/ML models to improve its decisions
  - E.g. Tests multiple configurations using a NN model able to predict the Wi-Fi throughput, and choses the one that is best performing.
- and is the final responsible of the obtained outcome.



- Make predictions
- Select most likely option
- Identify patterns
- …

Standardization: IEEE 802.11

Artificial Intelligence Machine Learning (AIML) Topic Interest Group (TIG)

- AIML Use Cases and Features for WLAN
  - AIML-based CSI feedback compression
    - Unsupervised learning allows to reduce CSI feedback
  - Deep-learning based distributed channel access
    - Deep learning allows to increase throughput and decrease latency (e.g., CW optimization)
  - Efficient AIML model sharing/distribution
    - Efficient model sharing needs to be defined
  - AIML Enhanced Roaming
    - AIML can improve roaming decisions
  - AIML based Multi-AP Transmission
    - AIML can help in finding AP-STA pairs for multi-AP transmissions, etc.

All AIML TIG documents available @ https://mentor.ieee.org/802.11/documents?is_dcn=aiml

Standardization: IEEE 802.11

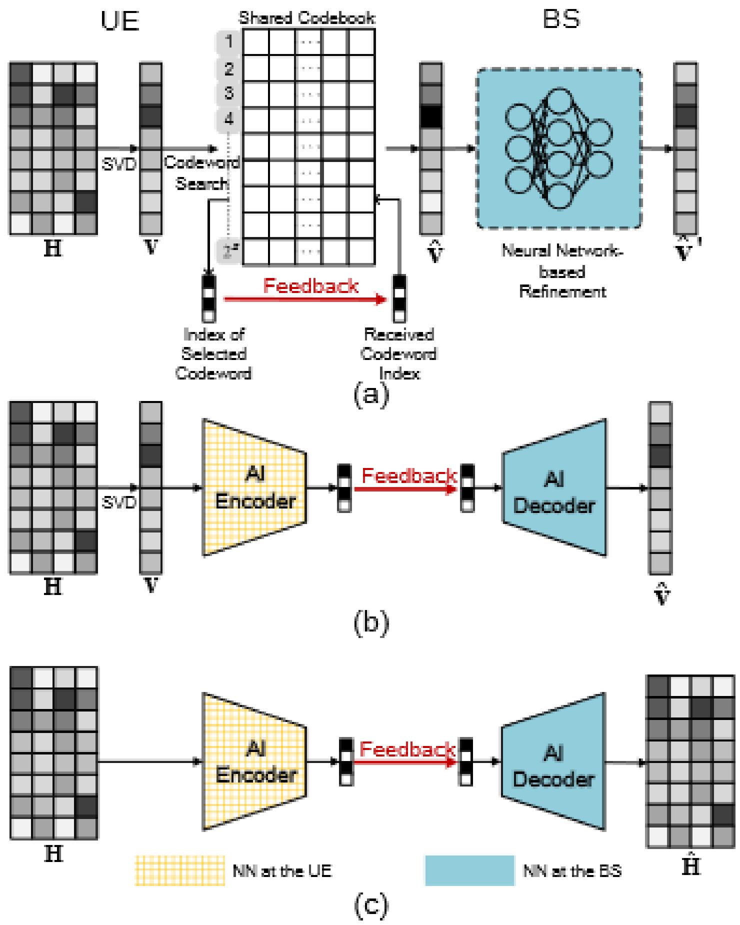
AIML-based CSI Feedback Compression

Problem

- In 802.11ax/be the AP uses a sounding sequence for beamformee/beamformer communication
- Sounding feedback airtime overhead increases with new features (e.g., MAP + many spatial streams)

Benefit from ML

- CSI compression
- Faster operation
- Improved throughput



Guo, Jiajia, et al. "AI for CSI feedback enhancement in 5G-advanced." *IEEE Wireless Communications* (2022).

Standardization: IEEE 802.11

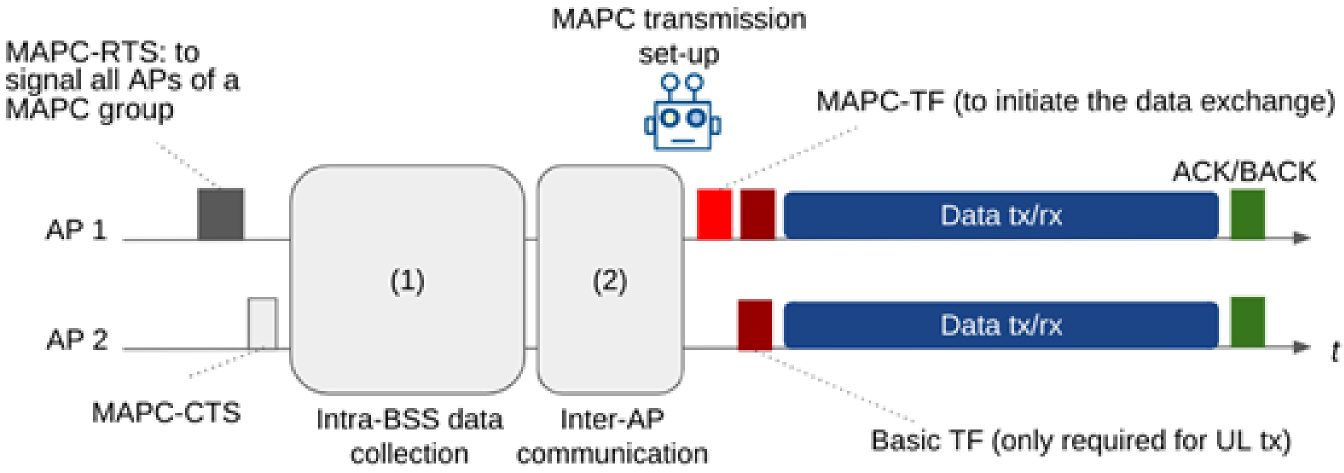
AIML-based Multi-AP Coordination

Problem

- Dense multi-AP networks suffer from co-channel interference, multi-AP coordination (MAPC) becomes necessary

Benefit from ML

- Selection of non-colliding groups → improved utilization of radio resources



Outline

Part 1: **Introduction. Why Wi-Fi may want to adopt AI/ML?** (30 mins) - Katarzyna

- Wi-Fi overview: A 30-year path from IEEE 802.11b to IEEE 802.11be, and beyond.
- Open challenges in Wi-Fi: Dealing with complexity and uncertainty.
- Requirements of next-gen WiFi networks.

Part 2: **A primer on AI/ML** (45 min) - Szymon & Boris

- Concepts, definitions, and overview of the main ML types (supervised, unsupervised, and reinforcement learning), including Wi-Fi examples.
- Popular ML techniques and paradigms: deep learning, reinforcement learning, online learning, federated learning.
- Deployment options: architecture, data handling, marketplaces.

Break (10:00-10:30)

Part 3: **Multi-Armed Bandits for Responsive Wi-Fi networks** (45 mins) - Boris & Szymon

- Multi-Armed Bandits: exploitation-exploration trade-off; e-greedy, Thompson Sampling, UCB.
- Examples: Channel Selection; AP selection;
- Reinforced-lib + Example (MCS selection).

Part 4: **Wi-Fi & ML: Practical considerations** (45 mins) - Francesc

- Wi-Fi & ML: A two-sided relationship
- Adoption of ML in Wi-Fi
- Hands-on Exercise II: Predicting the performance of Wi-Fi through Deep Learning

Part 5: **Open challenges, future research directions, and summary** (15 mins) - Katarzyna

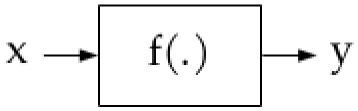
- Open challenges and future research directions; Synergies with other disruptive technologies.
- Current AI/ML trends: LLM and Generative AI.
- Summary and takeaways.

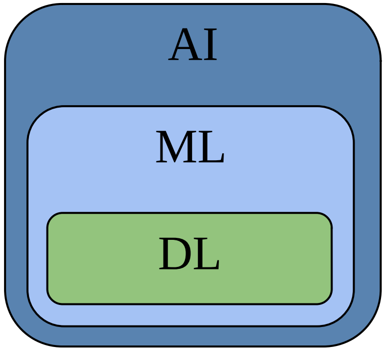
2
A Primer on AI/ML

Artificial Intelligence and Machine Learning

- **Artificial Intelligence**: collect/sense (get the data), reason (interpret the data), act (based on how the data is interpreted), enjoy (or not) the result, and self-correct / adapt.
  - AI uses ML techniques to extract "knowledge" (learn) from the data
  - Data + ML Algorithms + Reasoning + Actions→ (A)Intelligence
- **Machine Learning**: Set of algorithms / techniques / mechanisms that are able to learn from data.
  - **Learn**: map output to inputs, i.e., build the function f(.) that represents the 'knowledge'
  - General mechanisms, not explicitly programmed for a single family of functions f(.); In some cases, their performance/learning bounds are known
- **Deep Learning:** Includes learning the relevant 'features' of the model.

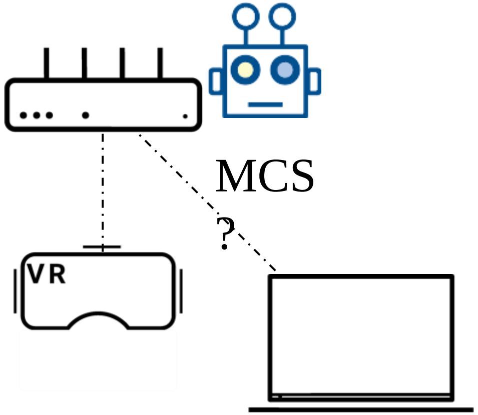
Intelligent agent:
body + policy (brain)



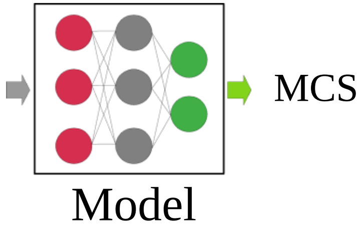


Building models

- The ML model is f(.), it describes a system.
- ML enables to model **extremely complex systems** by characterizing the hidden relationships between input parameters **x** (features) and outputs **y** (labels).
    - ML vs analytical models: ML models use ‘real data’
- **Example**: To train a model able to suggest the best MCS to exchange data between the AP and laptop.

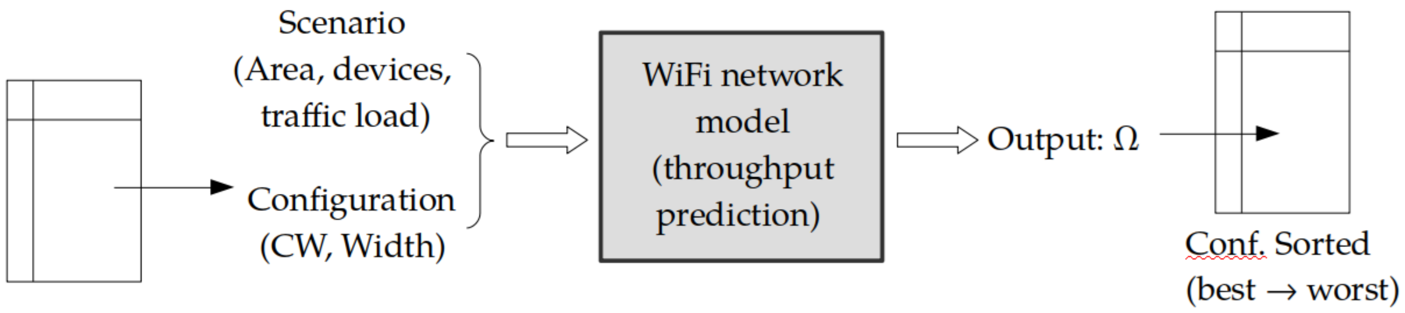


Features
- RSSI
- Throughput
- PER
- Coll. prob.
- SS
- Bandwidth



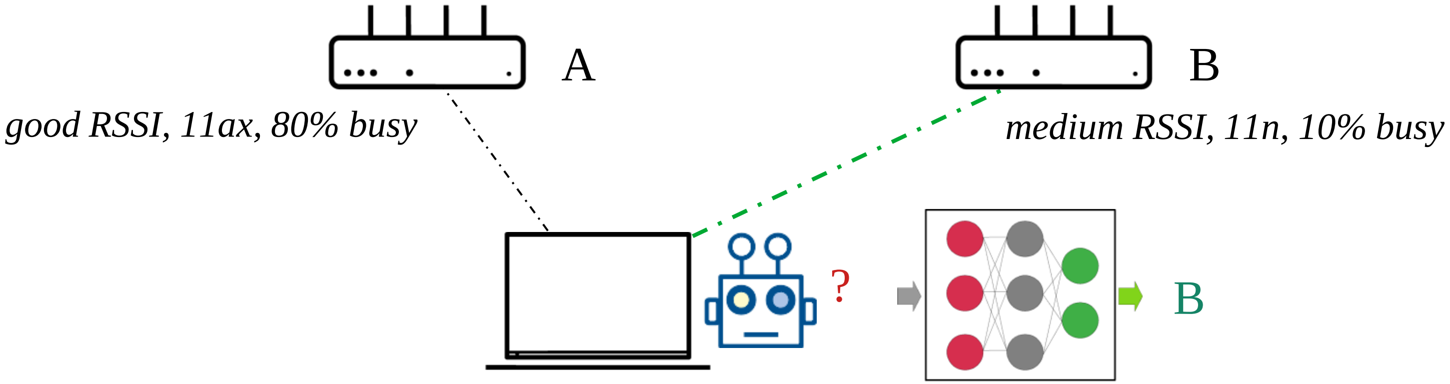
Why is having a model useful?

- A model can give us some useful information before taking (better) decisions.
- It can predict what will be the ‘performance’ (output) given a certain configuration (input).
- **Example 1**: Given a certain scenario, we want to find the configuration that maximizes the throughput



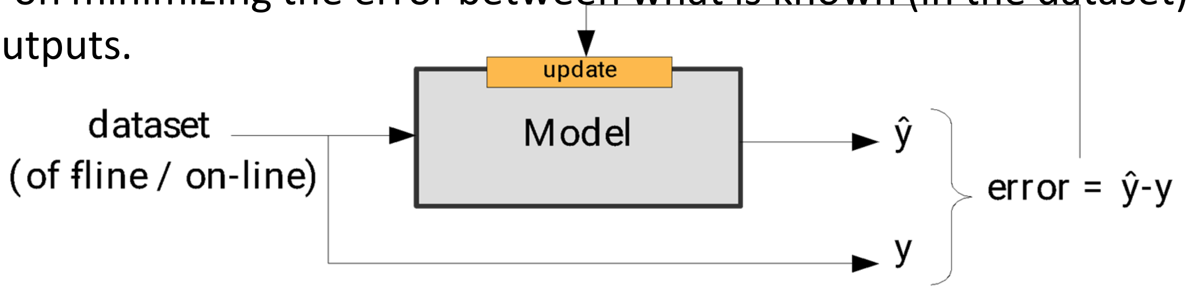
Why is having a model useful?

- A model can give us some useful information before taking (better) decisions.
- It can predict what will be the ‘performance’ (output) given a certain configuration (input).
- **Example 2**: After gathering data from different APs (load, number of associated clients, technology & capabilities, RSSI, etc.) a STA can sequentially predict the best AP to associate.



ML techniques build models from data (training)

- Two training cases:
  - **Off-line**: using a static dataset (present from the beginning).
  - **On-line**: the dataset is created in real-time, so the model is updated as more data is collected.
- Training is based on minimizing the error between what is known (in the dataset) and predicted (by the model) outputs.

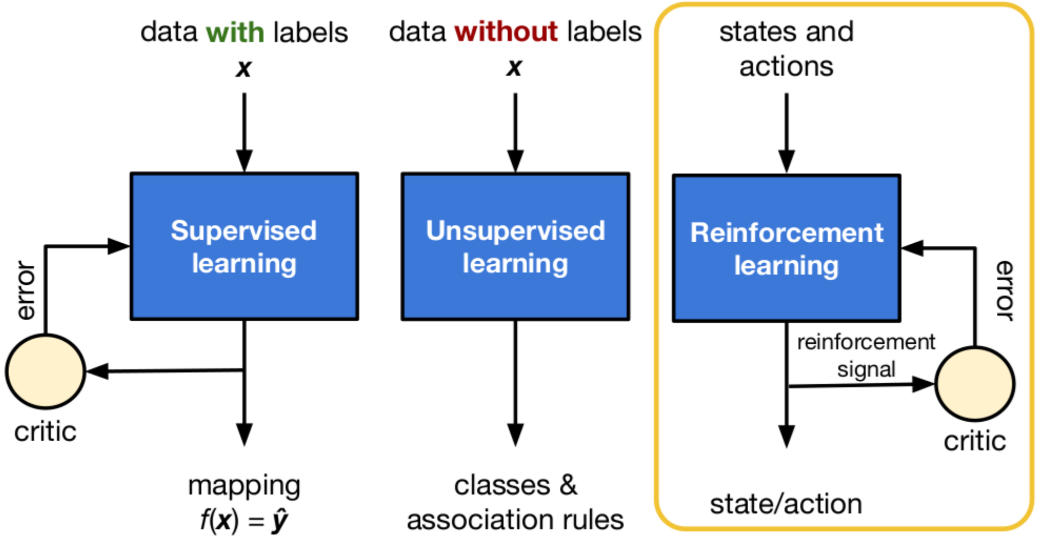


A model can be first trained off-line, deployed and then improved with on-line training

- Models should be as general as possible: the aim is not to exactly map inputs and outputs, but to learn the context behind any existing relationship (no overfitting).

Taxonomy of ML models

- Supervised Learning (SL)
- Unsupervised Learning (UL)
- Reinforcement Learning (RL)

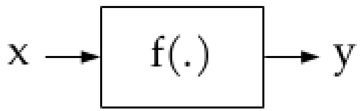


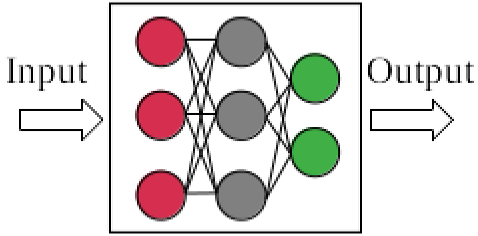
- How are these models trained?
  - Centralized; Decentralized / Distributed
  - Single player; Multiple players

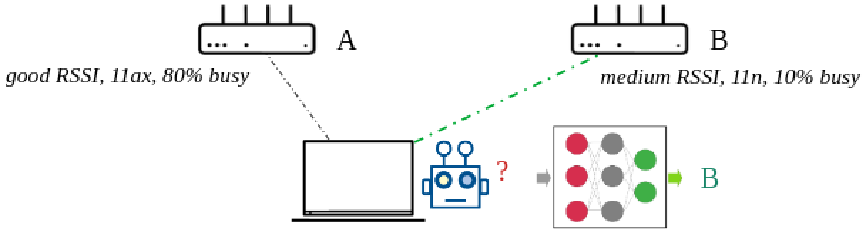
Ex: Distributed Federated Learning

Supervised Learning

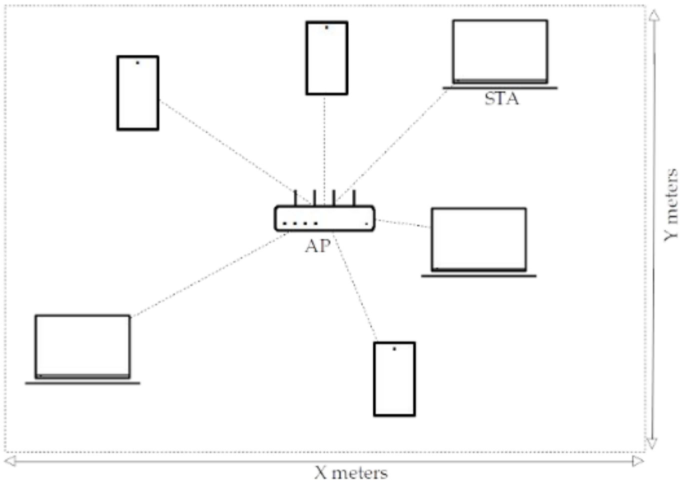
- **Goal**: Training data includes x and y, so the goal is to learn f(.)
- **Dataset**: We need a consolidated dataset to train the model, retraining as the dataset updates
- **Techniques**: Time series analysis, regressions, vector machines, decision trees, neural networks, etc.
- **Main uses:** Prediction (output is a real value), classification (output is a label, e.g., 0 or 1)
- **Wi-Fi example:** AP selection
  - The STA uses a trained model that predicts which is the best AP to associate. The input of the model should consist of the info gathered by the STA for each AP.







Performance Prediction with Linear Regression



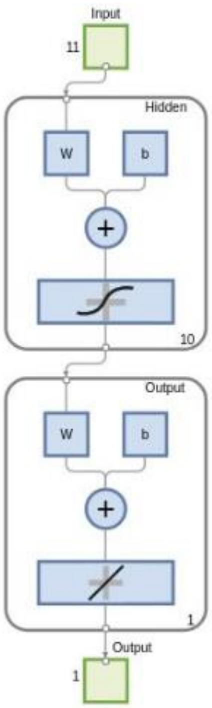
10k simulations → train model → validate model

Working example in Google Colab

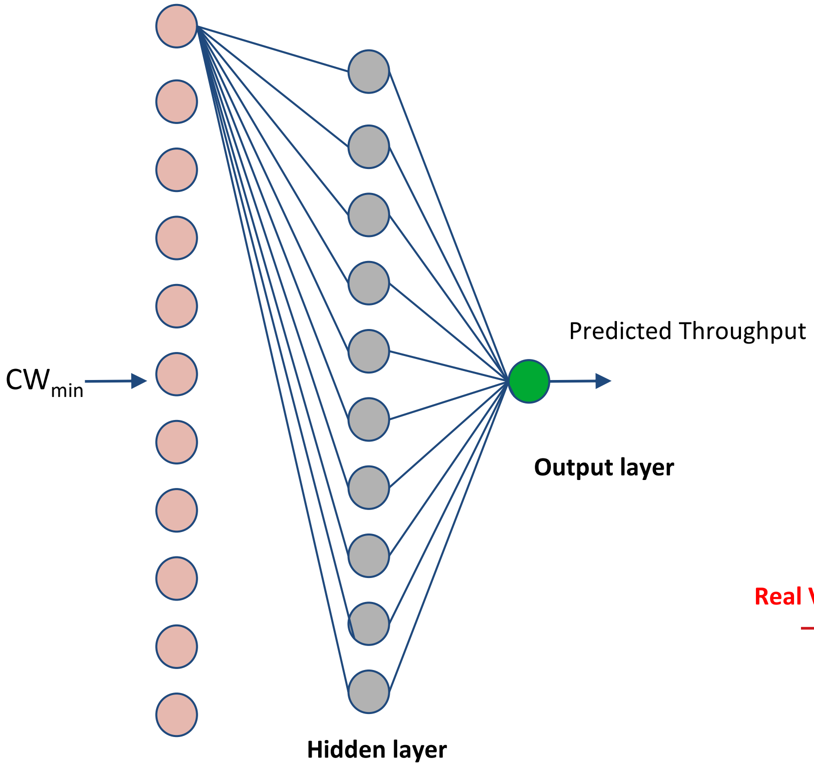
https://github.com/mlwifitutorial/mobicom2023/blob/main/Colab%20notebooks/Wi_Fi_Linear_Regression_Example.ipynb

Example: Neural Network

$x \rightarrow \boxed{f(.)} \rightarrow y$



Matlab NN representation



Train a neural network to map predictors to continuous responses.

Data

Predictors: input - [100000x11 double]

Responses: output - [100000x1 double]

input: double array of 100000 observations with 11 features.

output: double array of 100000 observations with 1 features.

Algorithm

Data division: Random

Training algorithm: Levenberg-Marquardt

Performance: Mean squared error

Training Results

Training start time: 03-Aug-2022 07:36:30

Layer size: 10

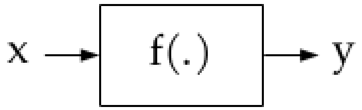
	Observations	MSE	R
Training	71000	0.1711	0.9965
Validation	15000	0.1692	0.9966
Test	14000	0.1685	0.9966

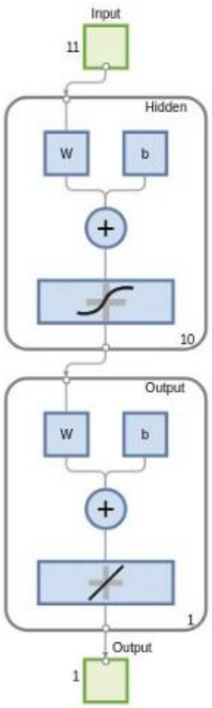
```
>> test = input(1294,:);
>> output(1294)

ans =

   6.7500e+00

>> y = NN_Throughput.Network(test')

y =

   6.0449e+00
```

Real Value (Mbps) →

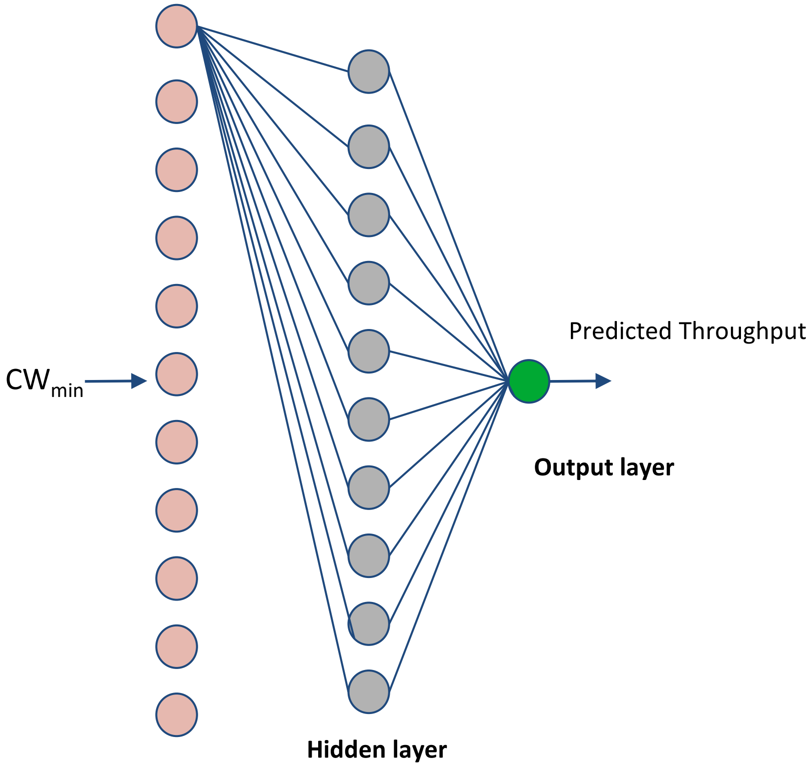
Prediction (Mbps) →

Example: Neural Network



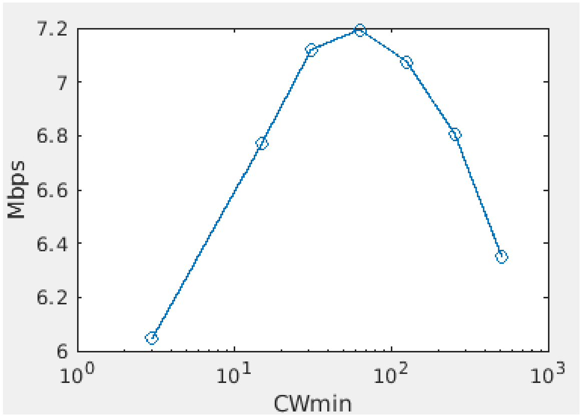


Matlab NN representation



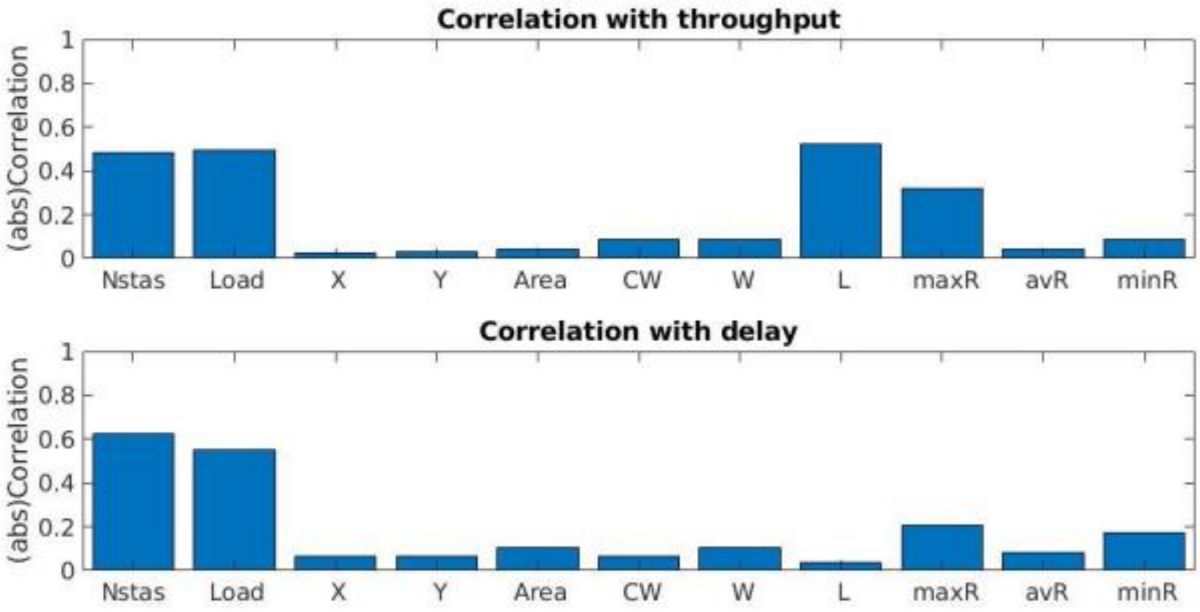
configuration to test

We can use the NN model to try different CW_{min} values to see which is the best performing one.



How to define the feature set?

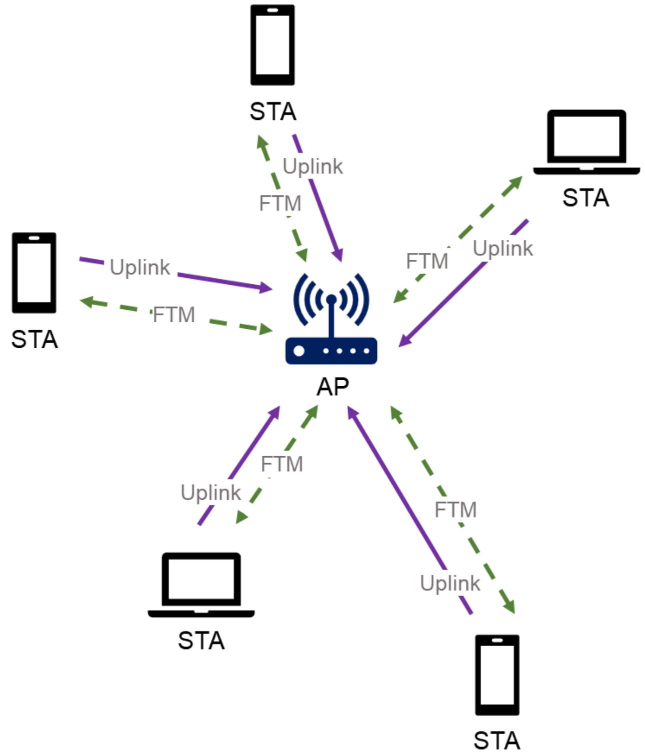
- We can study the cross-correlation between the desired output and the input features from the dataset, discarding those that show a low value.

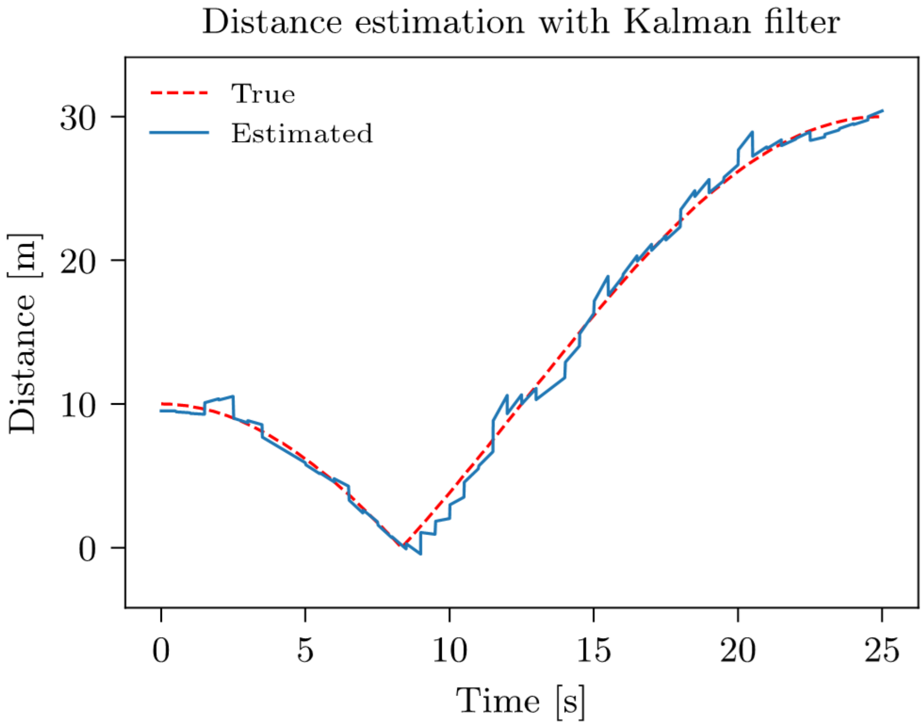


…or rely on deep learning solutions to choose the most appropriate ones.

FTMRate

https://github.com/ml4wifi-devs/ftmrate

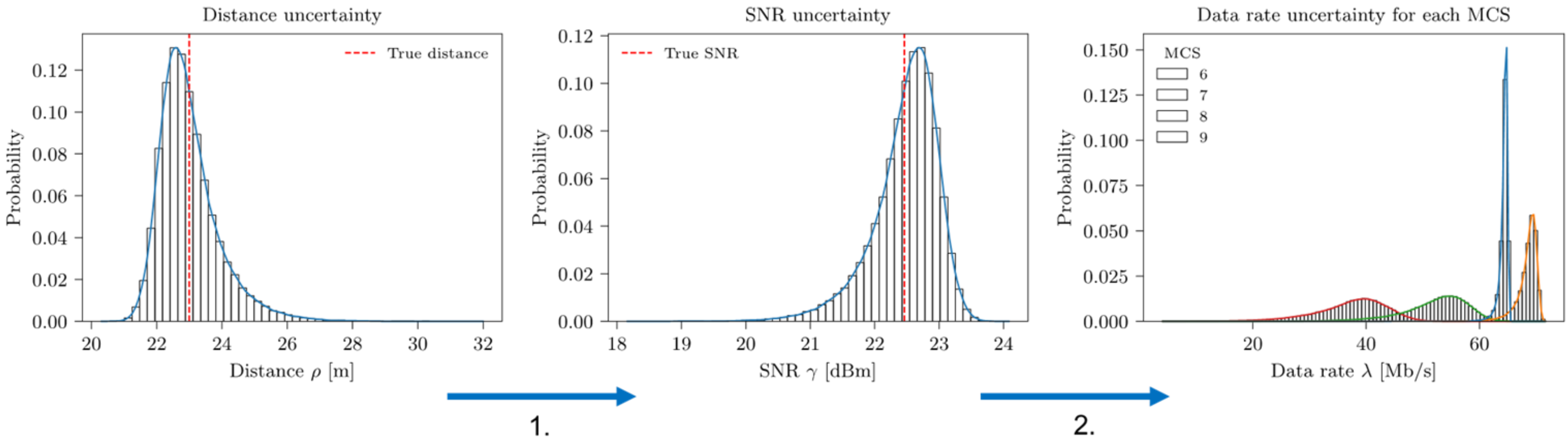




W. Ciezobka, M. Wojnar, K. Kosek-Szott, S. Szott, and K. Rusek. "FTMRate: Collision-Immune Distance-based Data Rate Selection for IEEE 802.11 Networks." 24th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2023.

FTMRate

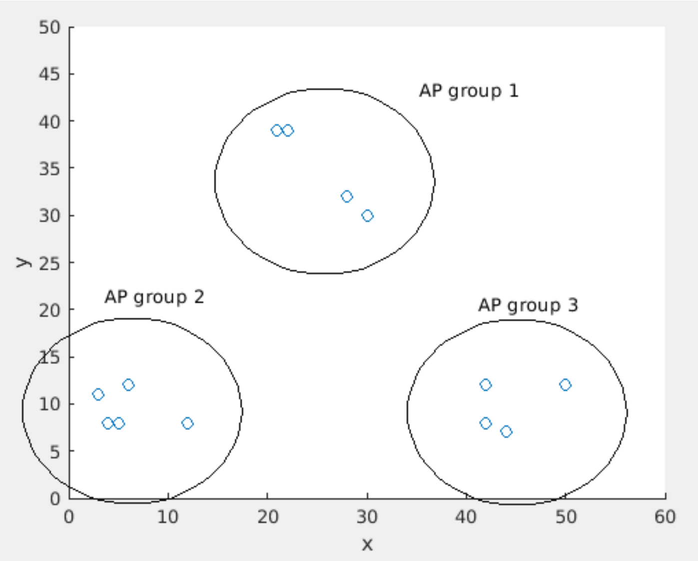
https://github.com/ml4wifi-devs/ftmrate



W. Ciezobka, M. Wojnar, K. Kosek-Szott, S. Szott, and K. Rusek. "FTMRate: Collision-Immune Distance-based Data Rate Selection for IEEE 802.11 Networks." 24th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2023.

Unsupervised Learning

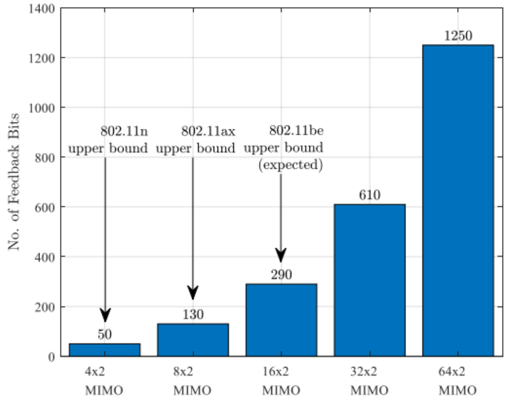
- **Goal**: Find patterns/trends/similarities in a dataset when only the features are available
- **Dataset**: We need a consolidated dataset to train the model, retraining it as the dataset updates
- **Techniques**: K-means, Self-Organizing Maps (SOMs), hidden Markov model (HMM), Auto encoders (AEs), Principal Component Analysis (PCA), Restricted Boltzmann machine (RBM), fuzzy C-means, etc.
- **Main uses:** Cluster the data in similar groups



- **Wi-Fi example**
  - AP selection from a dataset that contains only the features. Then, identifying which APs are similar may reduce the 'testing' phase by only requesting association to an AP of each representative 'cluster'.

Unsupervised Learning: CSI overhead reduction

- Source: Deshmukh, Mrugen, Mahmoud Kamel, Zinan Lin, Rui Yang, Hanqing Lou, and Ismail Güvenç. "**Intelligent Feedback Overhead Reduction (iFOR) in Wi-Fi 7 and Beyond**." In 2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring), pp. 1-5. IEEE, 2022.
- “Instead of feeding back the angles to represent one CSI vector, we propose the iFOR algorithm, which feeds back an index from a set of candidates that represent a diverse set of CSI feedback”
- Clustering N vectors in M (e.g., 1024, 10 bits): K-means.
- The M vectors are known at the AP and the stations.
- A station measures the CSI, and selects the candidate vector with a lower euclidian distance to the measured.
- Only the index (10 bits) is exchanged.



ıre 2 Comparing feedback overhead (in bits) for different MIMO configurations per subcarrier group.

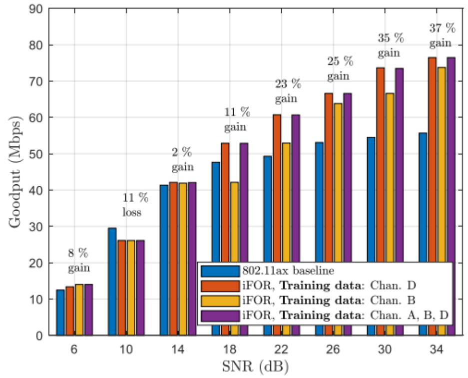
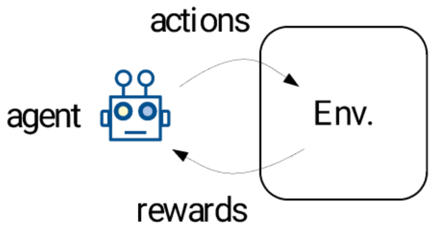


Figure 6 Goodput comparison for payload = 5000 bytes.

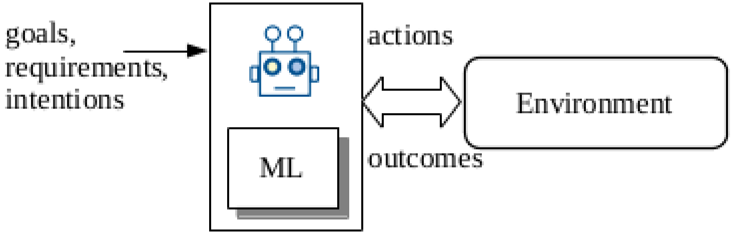
Reinforcement Learning

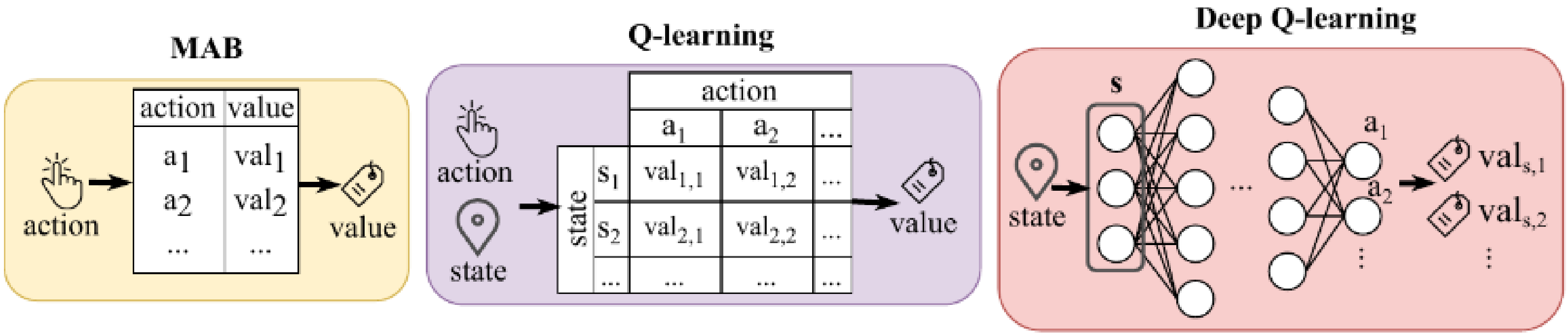
- **Goal**: Learn from experience (relationship between actions and rewards)
- **Dataset**: sequence of actions and rewards obtained
- **Techniques**: Multi-armed Bandits, Q-learning, DQL
- **Main uses:** Adaptive systems in dynamic environments
- **Wi-Fi examples: exploration – exploitation tradeoff**
  - AP selection by trying and rating different APs
  - Dynamic channel selection: given a set of unknown channels, we need to find the best one (the empty one)
  - Dynamic configuration of Tx Power and CCA to support Spatial Reuse
  - etc.



Reinforcement Learning

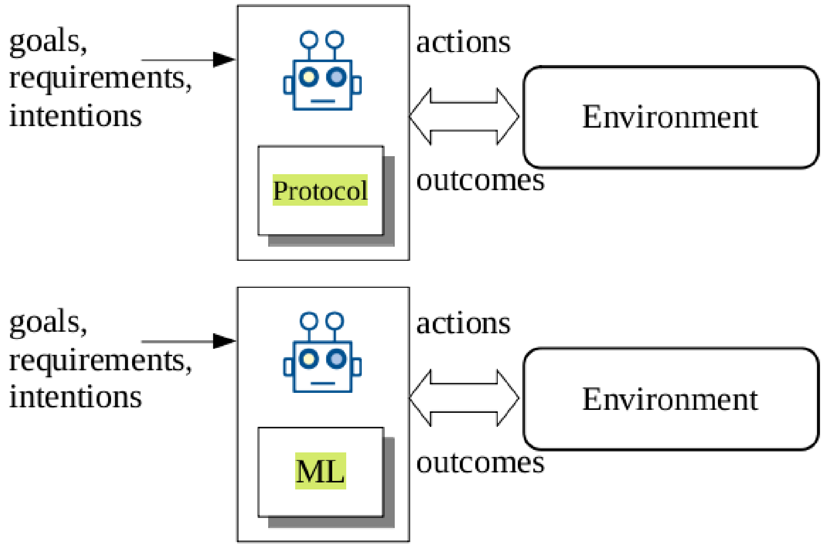
- An ‘**agent**’ learns by interacting with the ‘environment’
  - The agent can play different ‘actions’
  - For each action, the agent receives a ‘reward’
  - If possible, the agent may characterize the environment through a set of ‘states’
- The **ML algorithm** tells the agent how to select the actions to maximize the reward





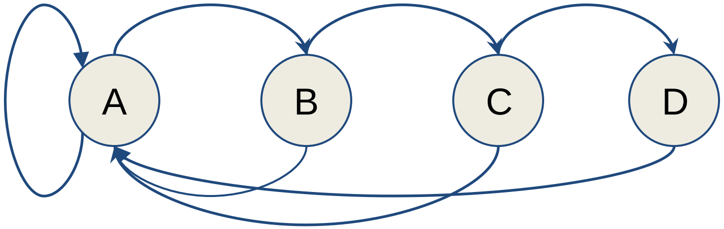
RL vs standard 'rule-based' protocols?

- RL naturally adapts to different situations.
- but…
  - … requires time to learn.
  - … higher computational resources.

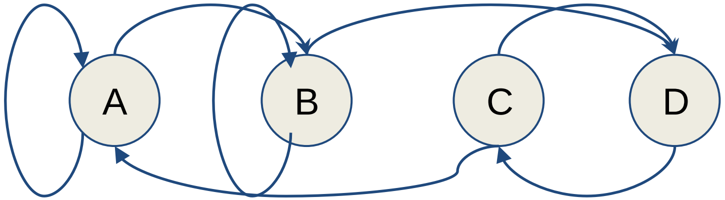


➔ The two approaches can be also combined.

Example: We can learn the protocol.

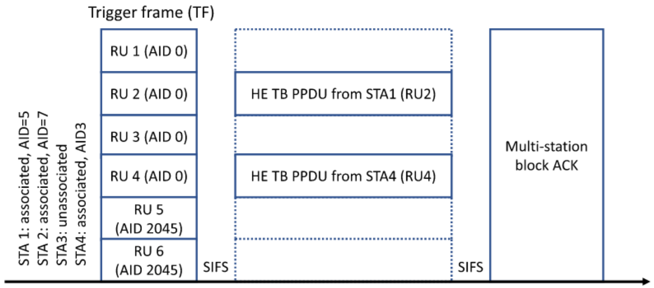


Pre-programmed

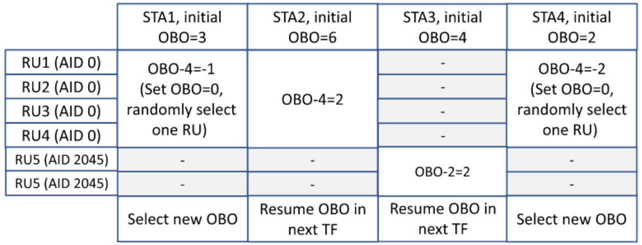


Learned

RL vs standard ‘rule-based’ protocols? 802.11ax Example: UORA



(a) UORA frame exchange



(b) OBO decrementation after TF reception

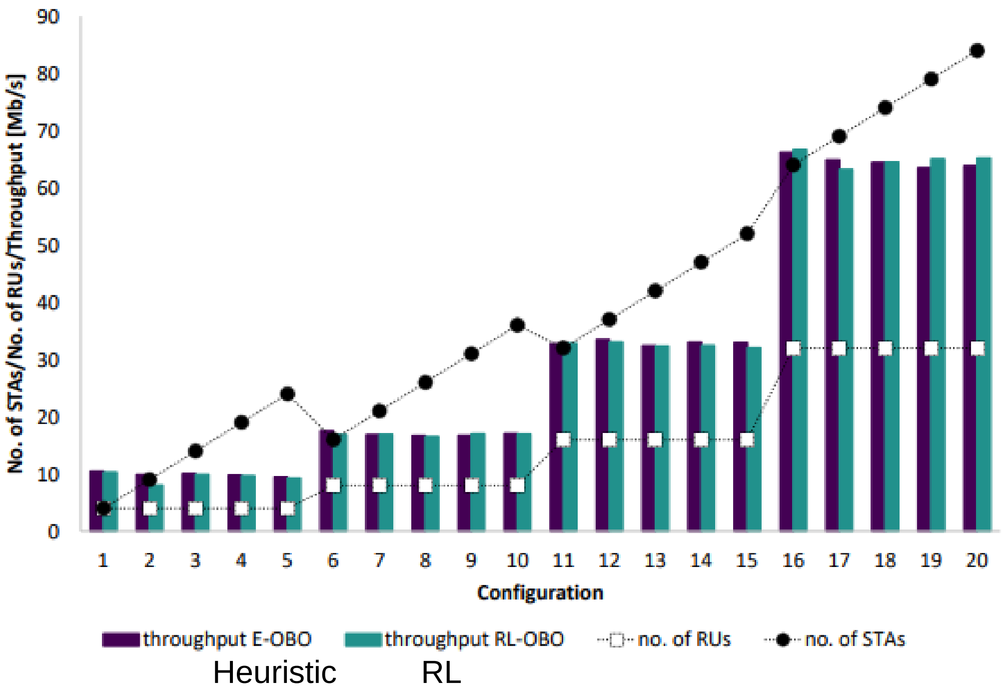
OBO decrementation

- Standard: fixed
- Efficient OBO: variable, depending on recent observations (heuristic) [1]
- RL-based OBO: variable, based on a trained DQL model [2]

[1] K. Kosek-Szott and K. Domino, ‘‘An efficient backoff procedure for IEEE 802.11ax uplink OFDMA-based random access,’’ IEEE Access, vol. 10, pp. 8855–8863, 2022
[2] Kosek-Szott, Katarzyna, Szymon Szott, and Falko Dressler. "Improving IEEE 802.11 ax UORA Performance: Comparison of Reinforcement Learning and Heuristic Approaches." IEEE Access 10 (2022): 120285-120295.

RL vs standard ‘rule-based’ protocols? UORA example

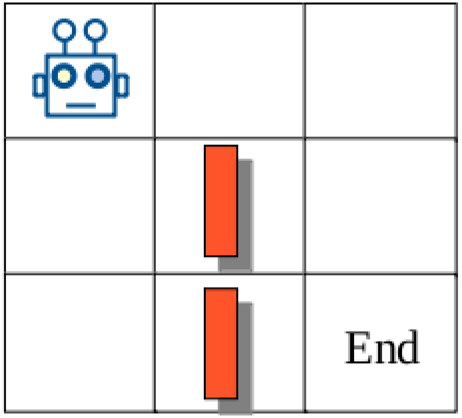
<u>Conclusions</u>: *“The simulation results confirm that the proposed approach gives satisfactory results in various dynamic settings; however, when compared to E-OBO, RL-OBO does no provide meaningful advantages. Both mechanisms provide similar outcome (i.e., throughput, channel access fairness, efficiency) and, therefore, th need for RL-OBO training and appropriate configuration of ML-related hyperparameters becomes a disadvantage.”*



Actions, Rewards and States

- Consider the case of a robot moving in a map.
- Each position of the map is a ‘**state**’: (x,y)
- **Actions**: the movements that can be done in each state: U, D, R, L
- **Rewards**: how close is the robot to the end; and/or if there is an obstacle.
- *Moving results in a reward, but also moves the system to a new state.*

➔ *Learning the model means knowing what to do (action to play) at every state.*

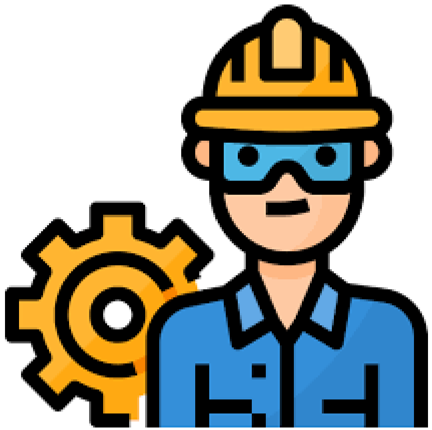


It is not clear how to define states in Wi-Fi, as sometimes the state is equivalent to the ‘action’ (i.e., the selected AP or channel).

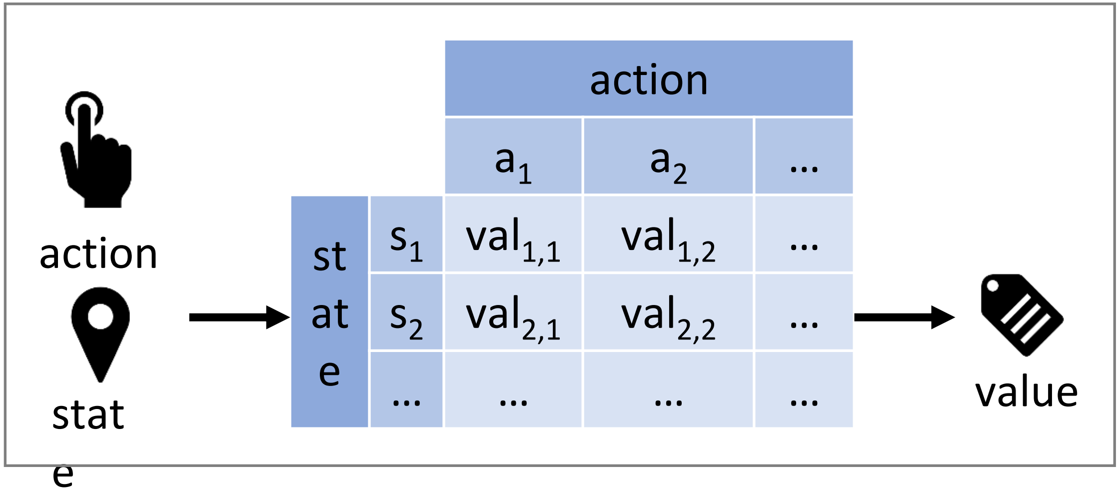
Wireless scenarios are hardly ‘static’, and so the same action may have different outcomes.

Designing Rewards

- A key aspect since it guides the behavior of the RL algorithm
- It can be a single value (last action), an average, the sum…
  - Last value: fluctuations;
  - Average: low response, as it depends on the past
- Q-learning rule: trade-off between past, current and future rewards
- Some algorithms require normalized rewards (i.e., [0,1])
- So?
  - For each problem, test different ones, and select the one working better.



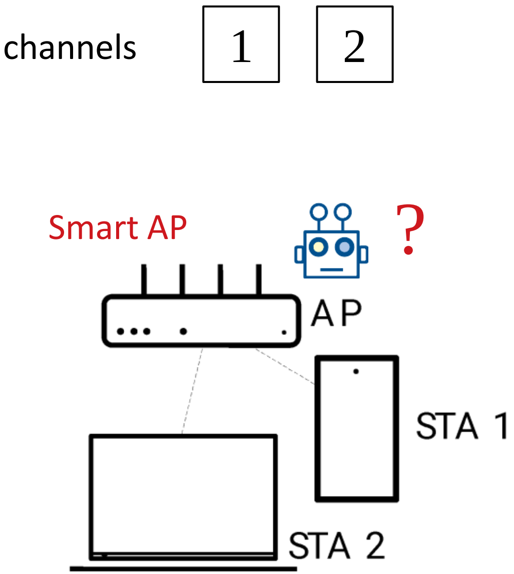
Q-learning

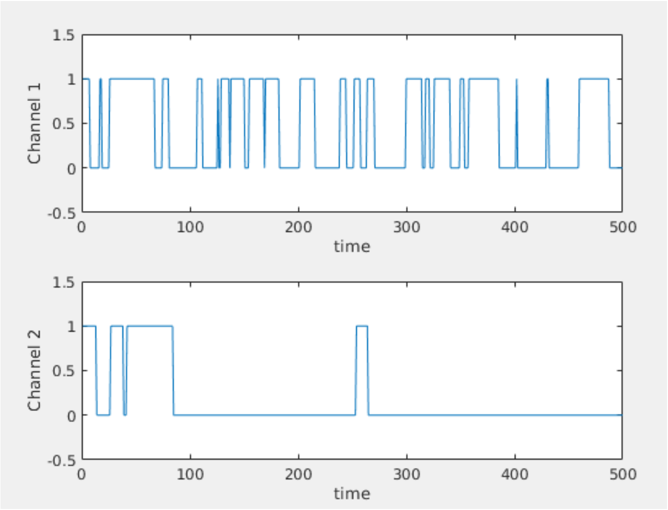


- It requires states:
  - Same or different actions depending on the state
  - Actions can result in a 'change' of state
  - Rewards depend on the state
- We need a mechanism to explore the environment
  - *e.g., ε-greedy*
- We need a mechanism to update Q-table based on the obtained rewards (Bellman eq.)

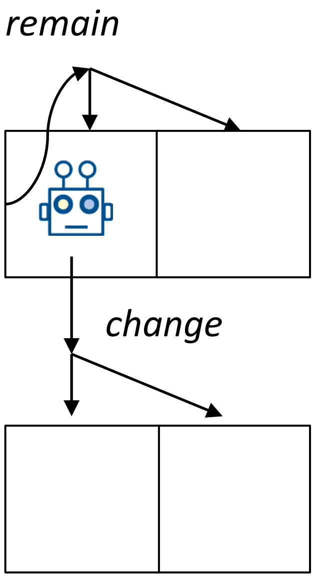
$$Q^{\text{new}}(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_a Q(s_{t+1}, a)\right)$$

Example: Channel Selection (Q-learning)





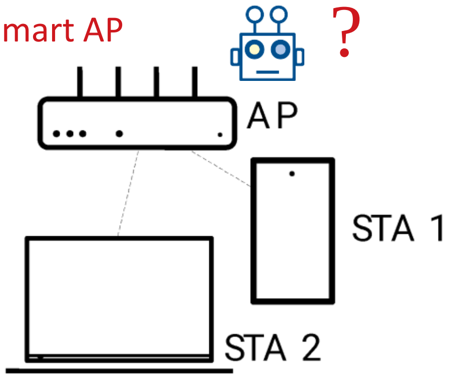
A channel can be busy or empty

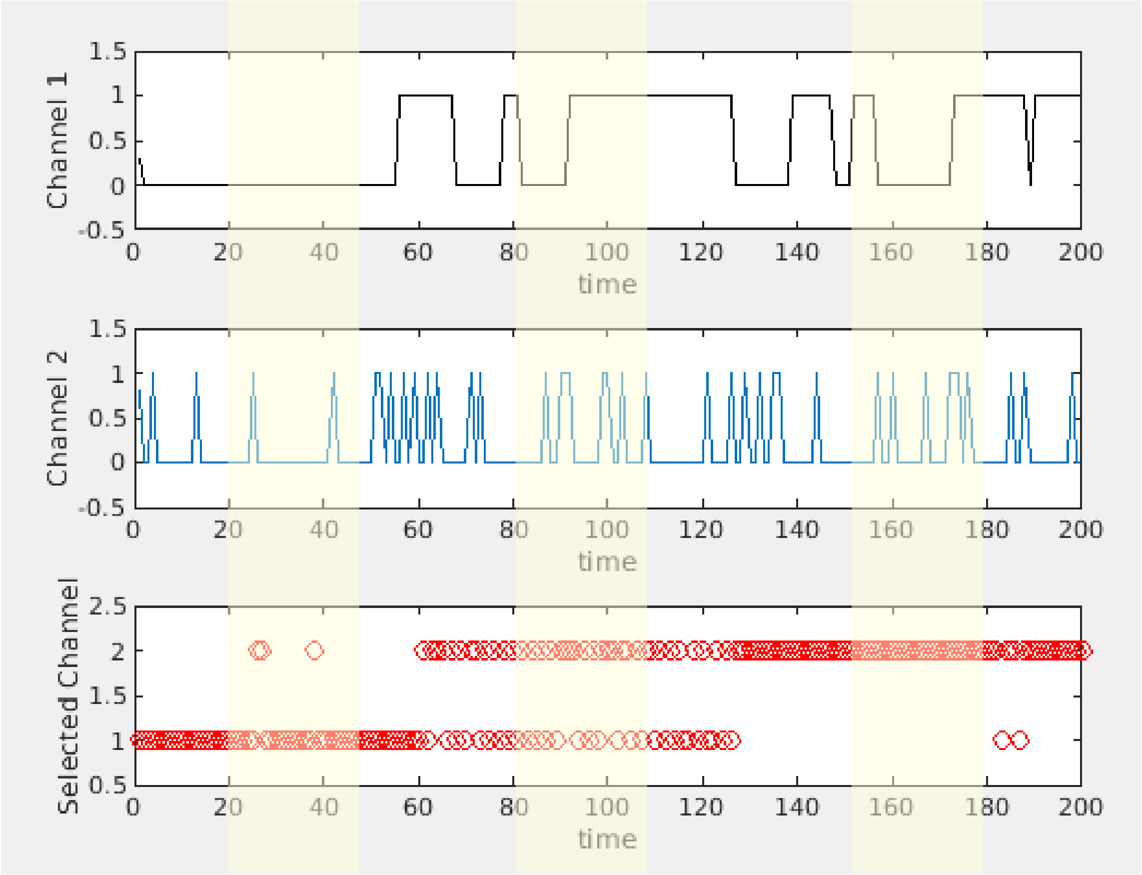


***Actions**: remain in the same channel; change channel*
***States**: channel 1 and empty; channel 1 and busy; channel 2 and empty; channel 2 and busy*

Example: Channel Selection (Q-learning)

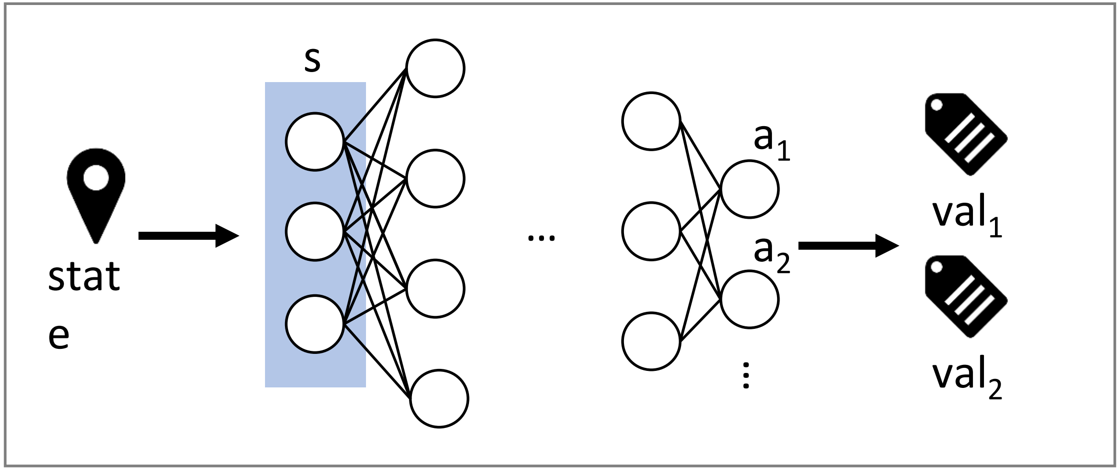
channels 1 2





a) b) c)

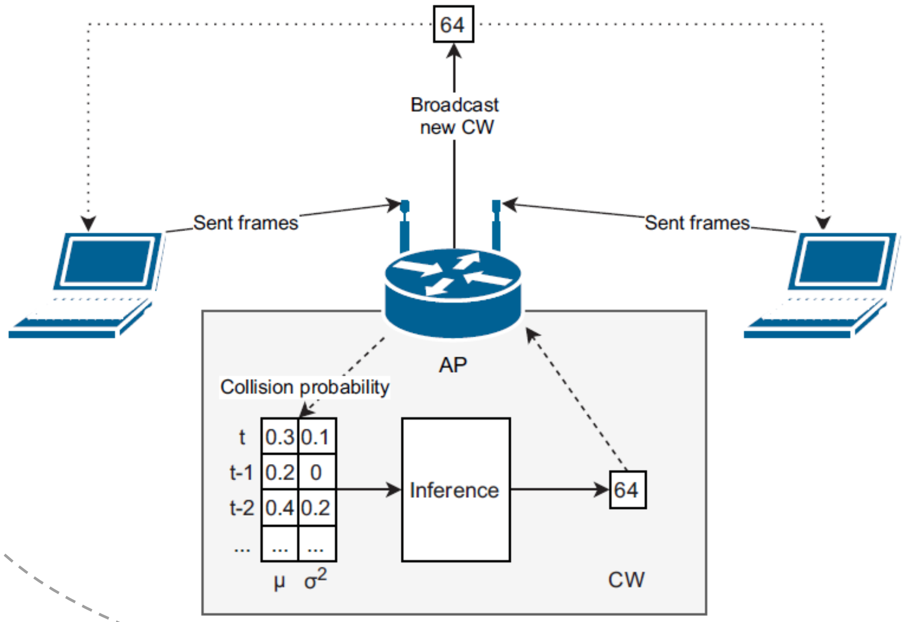
Deep Q-learning (DQN)



- **Problem**: Accurate learning of the Q-table is difficult (or impossible) and time consuming when the number of states is large as the agent needs to go through all the different states and actions several times.
- **Solution**: Model the Q-table response with a (Deep) Neural Network.
  - If the system is in state ‘s’, the agent asks the (D)NN to predict the reward the agent will obtain if it plays any of the available actions in state ‘s’.
  - The agent selects the action with a higher predicted ‘reward’.
  - DQN’s additional deep neural network allows for more efficient extrapolation of rewards **for yet unseen states** as compared to basic Q-learning.
- **Challenge**: to train on-line the (D)NN, as it may require large CPU/GPU resources.

Example: DQN for CW optimization

- Agent - installed at the AP
- State - status of all connected devices
- Action - select CW (value of integer *a*)
- Reward - network throughput
- Observation - collision probability
- DNN: 8 × 128 × 64
- Exploration: random action w/ low probability



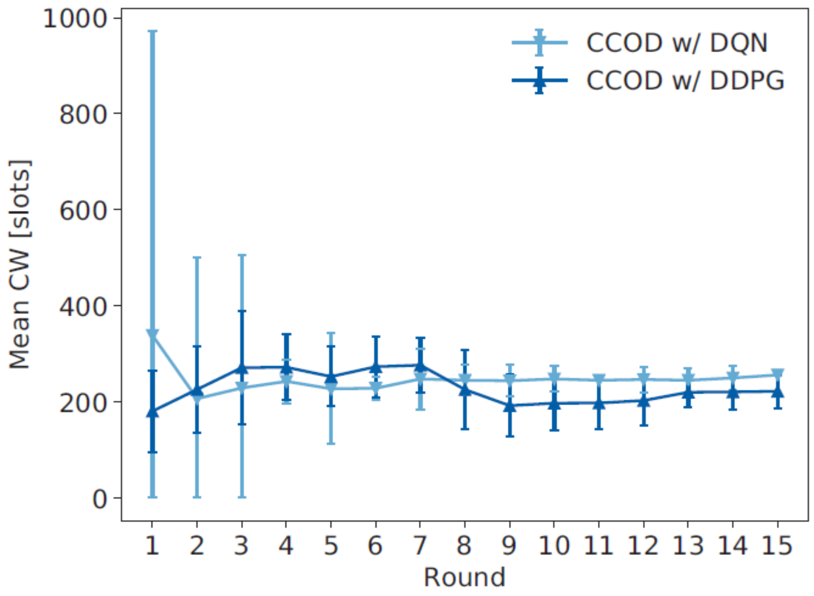
$$CW = \lfloor 2^{a+4} \rfloor - 1$$

$$a \in [0, 6]$$

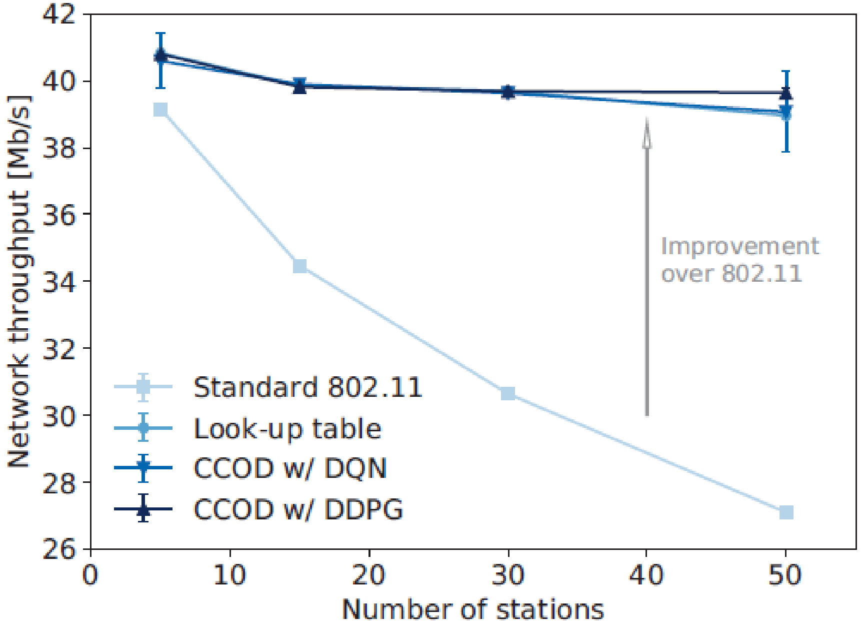
Wydmański, Witold, and Szymon Szott. "**Contention window optimization in IEEE 802.11ax networks with deep reinforcement learning.**" In 2021 IEEE Wireless Communications and Networking Conference (WCNC).

Example cont'd

- DQN - Discrete action space
- DDPG - Continuous action space



Mean selected CW in consecutive training rounds
(static scenario: 30 stations)



Throughput improvement over 802.11

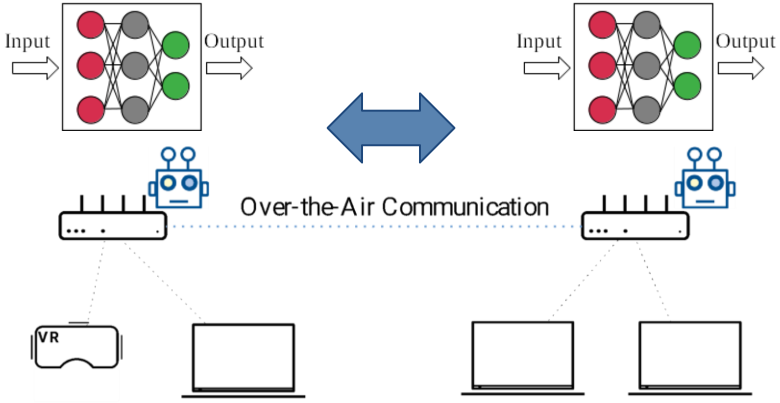
Federated Learning

- **Federated Learning**: devices exchange their locally trained 'global' models.
  - Centralized or distributed.
  - The 'global' model is re-trained locally using the data gathered by each device.
  - Only the 'model' is shared by participating devices. Data is kept local (privacy advantages + saves overheads).
    - SL: the weights of a NN
    - RL: Q-table, instantaneous rewards
  - The global model aggregates the weights from the local updates computed by clients (typically, FedAvg is adopted)
  - *Could improve system stability in 'adversarial' situations*

- **Example / Proposal**
  - To leverage the over the air MAPC framework to exchange local models to support DFL in next-gen. Wi-Fi networks

Algorithm 1 Federated Averaging (FedAvg)

for $t = 1, 2, \dots, T$ **do**
  for $k \in \mathbb{K}_{tr}$ **do in parallel**
    Pull $\boldsymbol{w}_t$ from central server: $\boldsymbol{w}_{t,0}^{(k)} = \boldsymbol{w}_t$
    for $e = 1, \dots, E$ **do**
      Update model: $\boldsymbol{w}_{t,e}^{(k)} = \boldsymbol{w}_{t,e}^{(k)} - \eta_t \nabla l_{t,e}^{(k)}$
    end for
    Push $\boldsymbol{w}_{t+1}^{(k)} \leftarrow \boldsymbol{w}_{t,E}^{(k)}$
  end for
  FedAvg: $\boldsymbol{w}_{t+1} = \frac{1}{|\mathbb{K}_{tr}|} \sum_{k \in \mathbb{K}_{tr}} \boldsymbol{w}_{t+1}^{(k)}$
end for



Transfer Learning: an example

- <u>Source</u>: Iturria-Rivera, Pedro Enrique, Marcel Chenier, Bernard Herscovici, Burak Kantarci, and Melike Erol-Kantarci. "**Cooperate or not Cooperate: Transfer Learning with Multi-Armed Bandit for Spatial Reuse in Wi-Fi**." *arXiv preprint arXiv:2211.15741* (2022).
- Each AP has an agent in charge of selecting the best TxPower and CCA threshold.
- Before selecting an action, the reward is predicted using a DNN, which is trained for each scenario (context)
- In dynamic scenarios, when there is a context change, the previous trained DNN is 'transferred' to the new one: a) completely, or b) partially (a subset of layers).

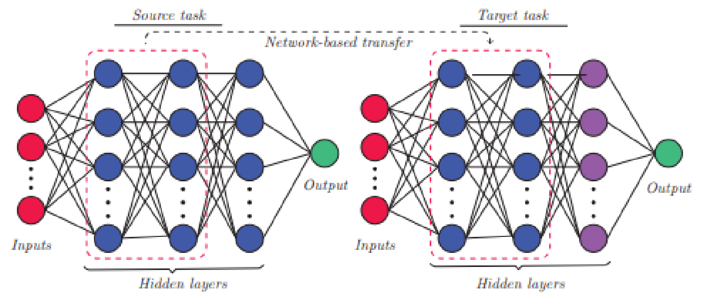
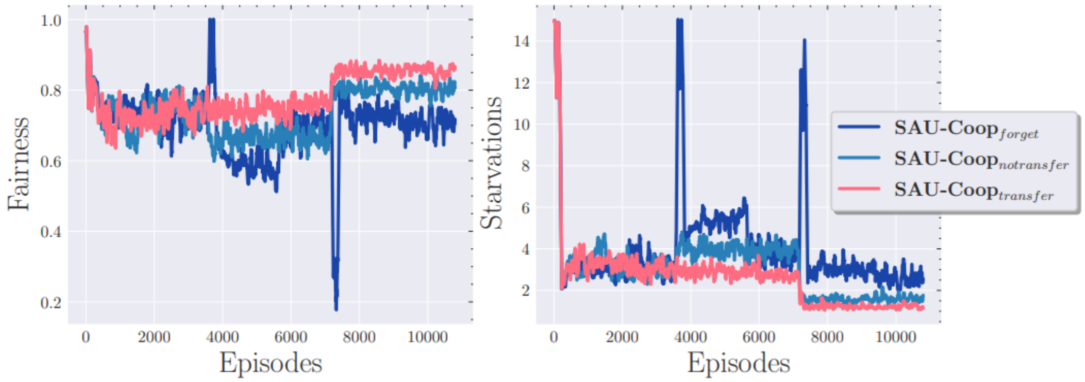


Fig. 2: Network-based transfer learning: the neural network source task's hidden layers are reutilized in the target network



Re-cap

- Machine Learning → techniques to create 'models' **from data**
  - <u>Supervised Learning</u>: Make predictions, classify
  - <u>Unsupervised Learning</u>: leverage similarity
  - <u>Reinforcement Learning</u>: interact with the environment
- Training and deploying models:
  - Off-line vs On-line
  - Centralized vs Distributed
- Use-cases / Examples:
  - Pre-trained SL models for AP selection (association)
  - RL for CW/channel selection in dynamic environments

The capability of ML to go beyond rule of thumb strategies by automatically learning (and adapting) to (un)seen situations can cope with heterogeneous wireless network scenarios

Outline

<u>Part 1</u>: **Introduction. Why Wi-Fi may want to adopt AI/ML?** (30 mins) - Katarzyna

- Wi-Fi overview: A 30-year path from IEEE 802.11b to IEEE 802.11be, and beyond.
- Open challenges in Wi-Fi: Dealing with complexity and uncertainty.
- Requirements of next-gen WiFi networks.

<u>Part 2</u>: **A primer on AI/ML** (45 min) - Szymon & Boris

- Concepts, definitions, and overview of the main ML types (supervised, unsupervised, and reinforcement learning), including Wi-Fi examples.
- Popular ML techniques and paradigms: deep learning, reinforcement learning, online learning, federated learning.
- Deployment options: architecture, data handling, marketplaces.

Break (10:00-10:30)

<u>Part 3</u>: **Multi-Armed Bandits for Responsive Wi-Fi networks** (45 mins) - Boris & Szymon

- Multi-Armed Bandits: exploitation-exploration trade-off; e-greedy, Thompson Sampling, UCB.
- Examples: Channel Selection; AP selection;
- Reinforced-lib + Example (MCS selection).

<u>Part 4</u>: **Wi-Fi & ML: Practical considerations** (45 mins) - Francesc

- Wi-Fi & ML: A two-sided relationship
- Adoption of ML in Wi-Fi
- Hands-on Exercise II: Predicting the performance of Wi-Fi through Deep Learning

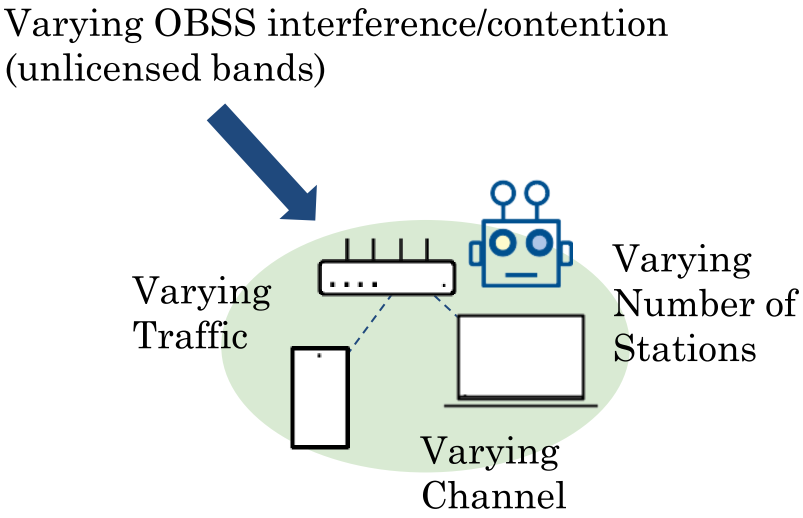
<u>Part 5</u>: **Open challenges, future research directions, and summary** (15 mins) - Katarzyna

- Open challenges and future research directions; Synergies with other disruptive technologies.
- Current AI/ML trends: LLM and Generative AI.
- Summary and takeaways.

3
Multi-Armed Bandits (MABs) for Responsive Wi-Fi networks

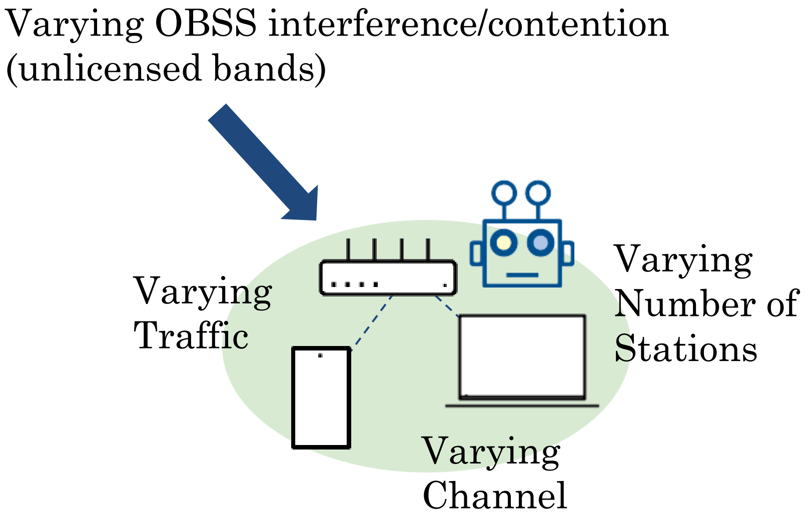
Motivation

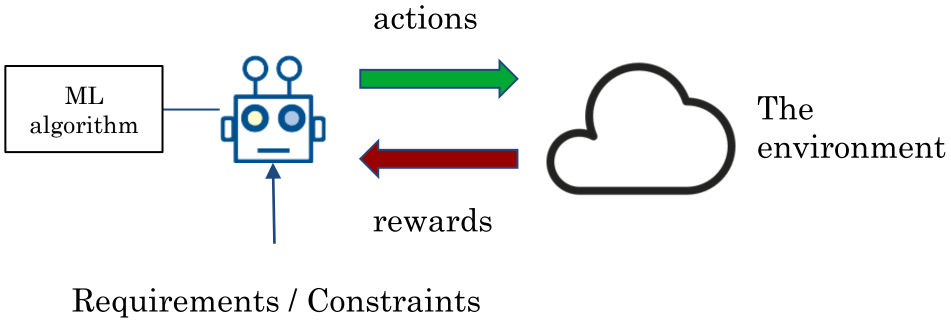
- There are multiple random phenomena that could affect Wi-Fi performance
  - Variable traffic
  - Variable channel conditions
  - Variable number of contenders
  - Variable OBSS interference/contention
- Environment changes fast: is it stationary or non-stationary?
  - Hard to find an 'optimal' configuration for all conditions.
- Can 'AI/ML' agents deployed in AP / stations help?
  - It will need to learn and react fast
  - One big 'agent' or multiple 'small agents'?



Motivation

The ML algorithm controls how the 'agent' explores (collects new information) and exploits what has already learnt



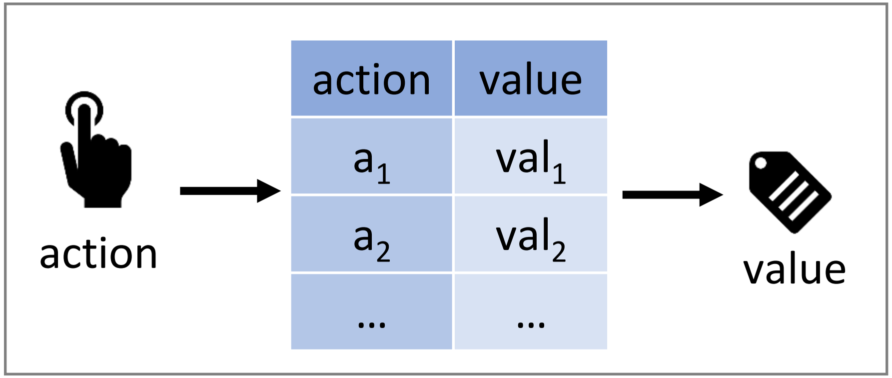


Goal: achieve the requirements using the minimum possible resources

Multi-armed Bandits (MABs) seem good candidates as light-weight ML algorithms to drive the exploration / exploitation trade-off

Multi-armed Bandits (MABs)

- Sequential interaction between the ‘agent’ and the environment:
  - action 1 → reward 1; action 2 → reward 2
- No states (or just a single state), only actions and rewards.
- Exploration-exploitation trade-off. The agent:
  - **Explores**: selects an action ‘randomly’
  - **Exploits**: selects the action with higher reward from the table
- Different approaches: simple, low-computing res.
  - *ε-greedy, Thompson Sampling, UCB, EXP3, etc.*
- **Contextual MABs** take into account different situations:
  - E.g. AP selection at home, at work
  - Moving to a new ‘context’ may allow to reuse past info.



Algorithm 1: Implementation of Thompson sampling.

Input : set of possible actions, $\mathcal{X} = \{x_1, ..., x_N\}$

Initialize: for each arm $x_i \in \mathcal{X}$, set $\hat{\mu}_i = 0$ and $n_i = 0$

while *active* **do**

  For each arm $x_i \in \mathcal{X}$, sample $\theta_i(t)$ from $\mathcal{N}(\hat{\mu}_i(t), \sigma_i^2(t))$

  Select arm $x_i = \underset{i=1,\dots,N}{\operatorname{argmax}}\ \theta_i(t)$

  Observe and compute the reward experienced $r_i(t)$

  $\hat{\mu}_i(t) \leftarrow \frac{\hat{\mu}_i(t) \cdot n_i(t) + r_i(t)}{n_i(t)+2}$

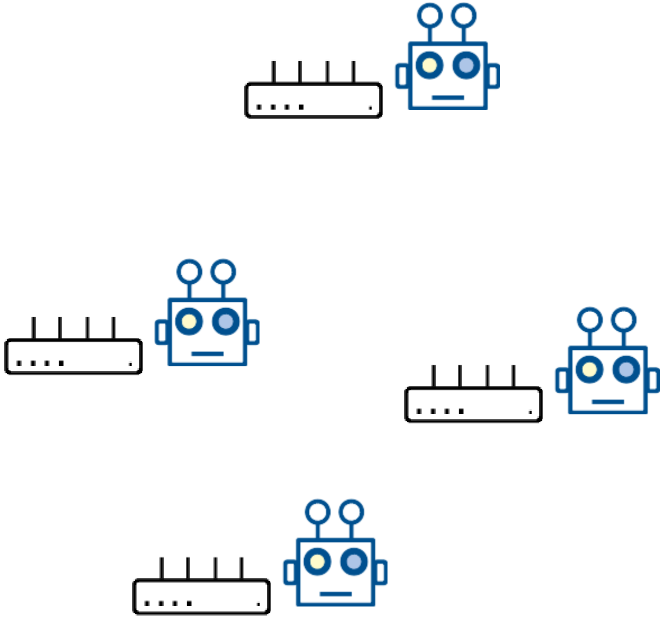
  $n_i(t) \leftarrow n_i(t) + 1$

end

Wilhelmi, F., Cano, C., Neu, G., Bellalta, B., Jonsson, A., & Barrachina-Muñoz, S. (2019). **Collaborative spatial reuse in wireless networks via selfish multi-armed bandits**. Ad Hoc Networks, 88, 129-141.

Single vs Multi-agent set-ups

- Multiple devices/networks trying to find a good configuration may result in a never-end story.
- Even if there is a joint action, it is unlikely that all agents choose the ‘right’ action at the ‘right’ moment.
- That optimal action for the entire network may be suboptimal for some individual ‘agents’, and so the equilibrium is fragile.
- To solve this problem,
  - temporal randomization can be applied (learn by turns): Agents learn only in certain periods of time. Idea: the best for me, it is the best for the others (not always true)
  - sharing information, and reaching consensus (federated learning)



Multi-armed Bandits (MABs) - To know more

Baseline:

- Katehakis, M. N., & Veinott Jr, A. F. (1987). The multi-armed bandit problem: decomposition and computation. Mathematics of Operations Research, 12(2), 262-268.
- Gittins, J., Glazebrook, K., & Weber, R. (2011). Multi-armed bandit allocation indices. John Wiley & Sons.
- Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2-3), 235-256.
- Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995, October). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on (pp. 322-331). IEEE.

Overviews:

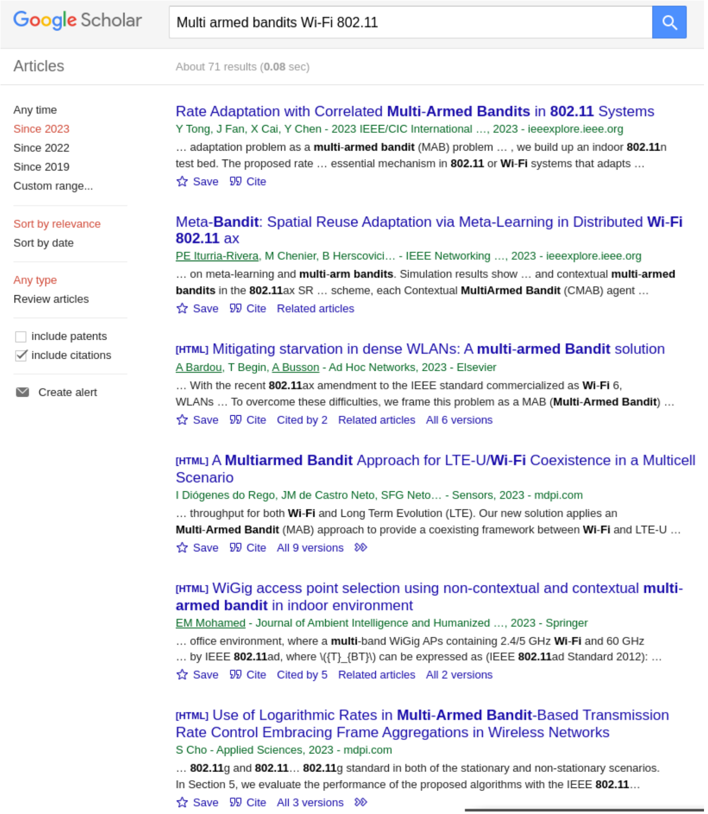
- Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning, 5(1), 1-122.
- Vermorel, J., & Mohri, M. (2005, October). Multi-armed bandit algorithms and empirical evaluation. In European conference on machine learning (pp. 437-448). Springer, Berlin, Heidelberg.

MAB Examples

- Channel selection
  - We introduce 3 basic MAB algorithms
  - S. Barrachina-Muñoz, et al.. "[Multi-armed bandits for spectrum allocation in multi-agent channel bonding WLANs]()." IEEE Access 9 (2021): 133472-133490.
- AP selection in multi-AP networks
  - A good lesson from multi-agent environments
  - M. Carrascosa, et al. "[Multi-armed bandits for decentralized AP selection in enterprise WLANs]()." Computer Communications 159 (2020): 108-123.
- Reinforced-lib – universal RL library
  - Analysis of MABs for rate selection: ε-greedy, UCB, TS, EXP3, Softmax

Is the research community using MABs?

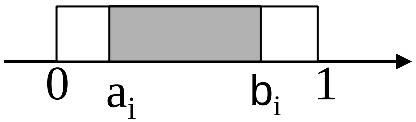
- Check our survey! Many papers using MABs.
- Some recent ones (Google Scholar)



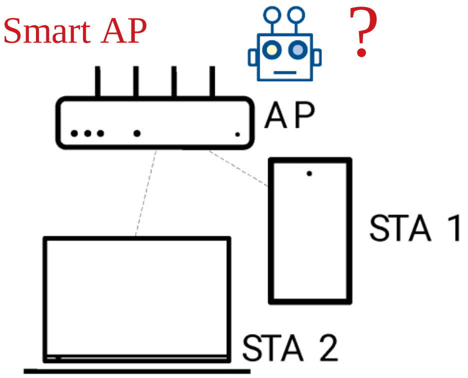
Since 2019: About 457 results (0.07 sec)

Example: Channel Selection (MABs)

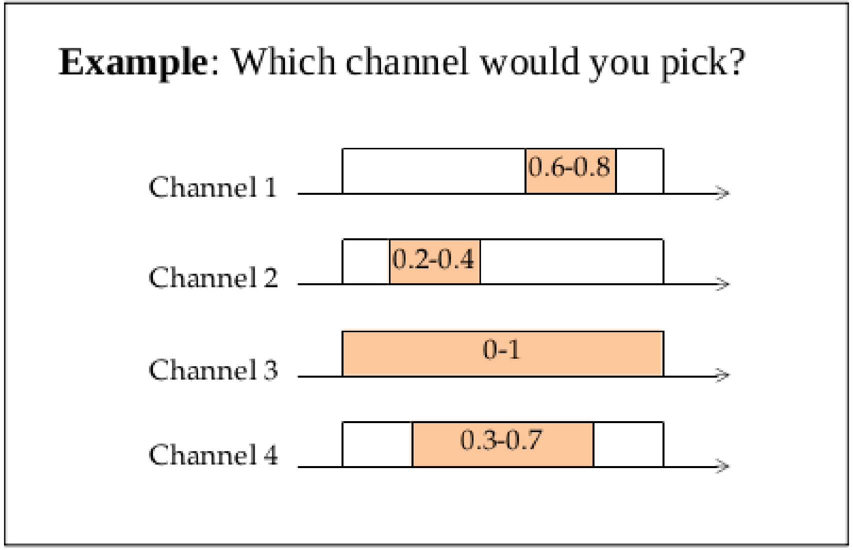
channels 1 2 3 .... N



ρ(i) = Occupancy of channel i [0,1]
(measured every 1 second)



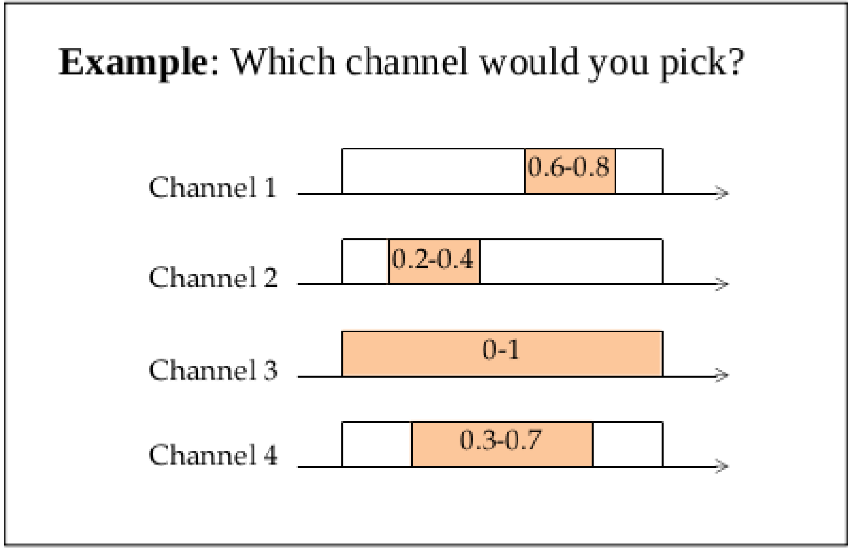
Reward using channel i: 1-ρ(i)

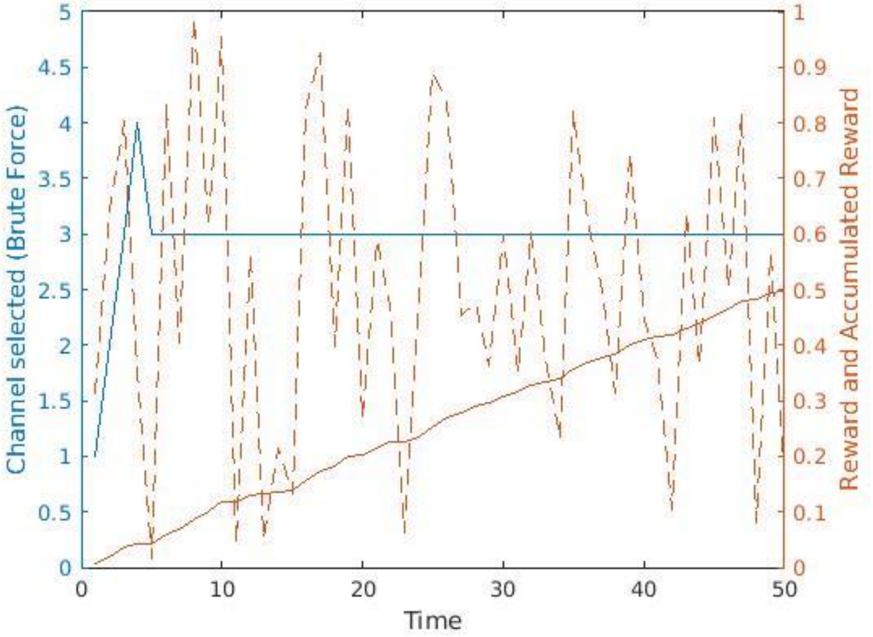


Only the channel in use can be measured

Sequential Exploration (Brute Force)

- We check sequentially all channels, and select the 'best' one for the rest of the time.





ε-greedy

- Direct implementation of the Exploitation / Exploration trade-off
  - Explore with probability ε
  - Exploit with probability 1-ε
- Performance of ε-greedy can be improved by:

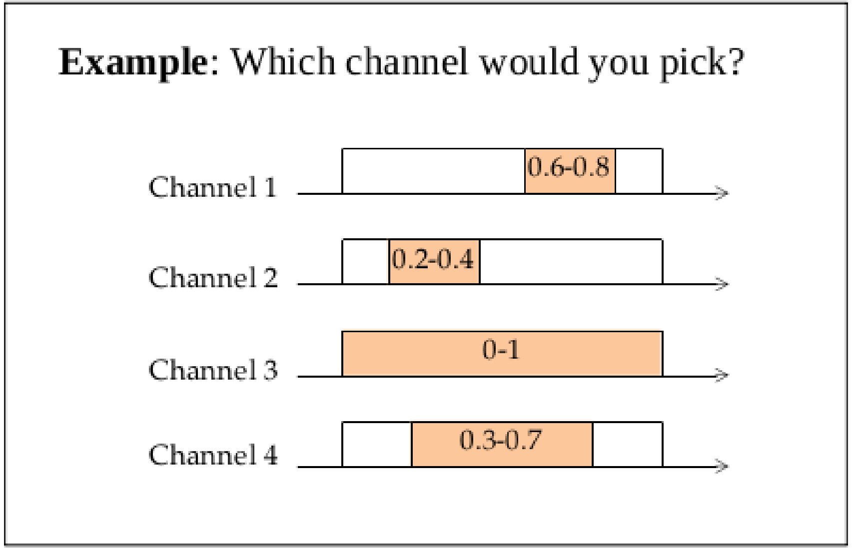
  [*Not a general rule, depends on the problem*]
  - Updating the value of ε as the system is learning
  - Adjusting how the average reward for every action is selected (i.e., weighted average)

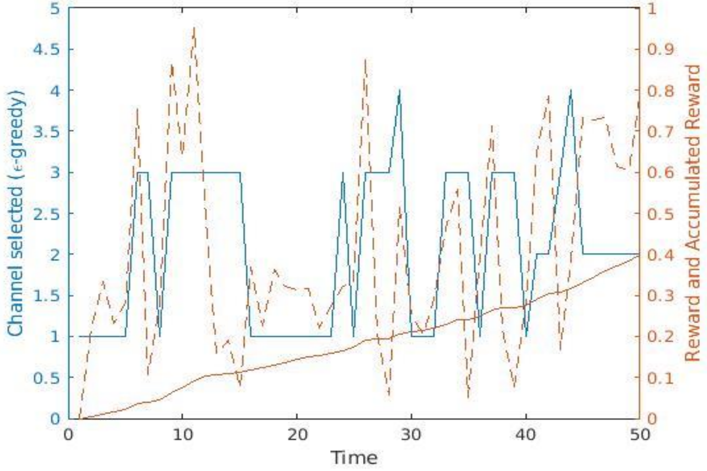
Algorithm 1: Multi-Armed Bandits (ε-greedy)

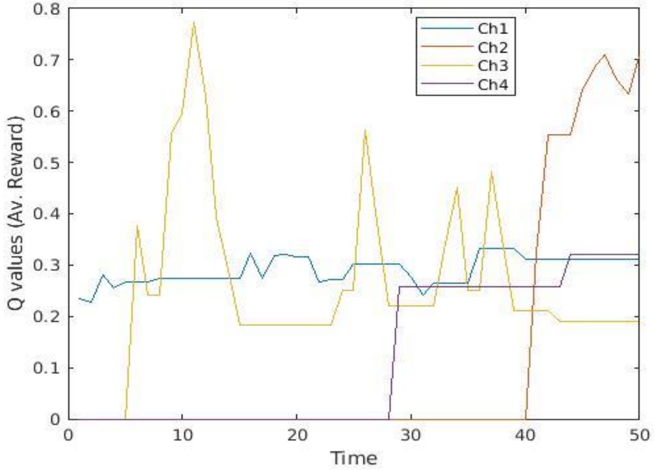
Input: $\mathcal{A}$: set of possible actions $\{a_1, ..., a_K\}$
Initialize: $t = 1$, $\varepsilon_t = \varepsilon_0$, $r_{k,0} = 0, \forall a_k \in \mathcal{A}$
while *active* **do**
  Select $a_k \begin{cases} \underset{k=1,\dots,K}{\text{argmax}}\ r_{k,t-1}, & \text{with prob. } 1-\varepsilon \\ k \sim \mathcal{U}(1,K), & \text{otherwise} \end{cases}$
  Observe the channel occupancy $\rho_{k,t}$
  Compute the reward $r_{k,t} = (1-\alpha)r_{k,t-1} + \alpha(1-\rho_{k,t})$
  $\varepsilon_t \leftarrow \varepsilon_0/\sqrt{t}$
  $t \leftarrow t+1$
end

ε-greedy

- N=4 channels
- 1 episode, 50 iterations
- α=0.5; ε_0=0.2 (constant)

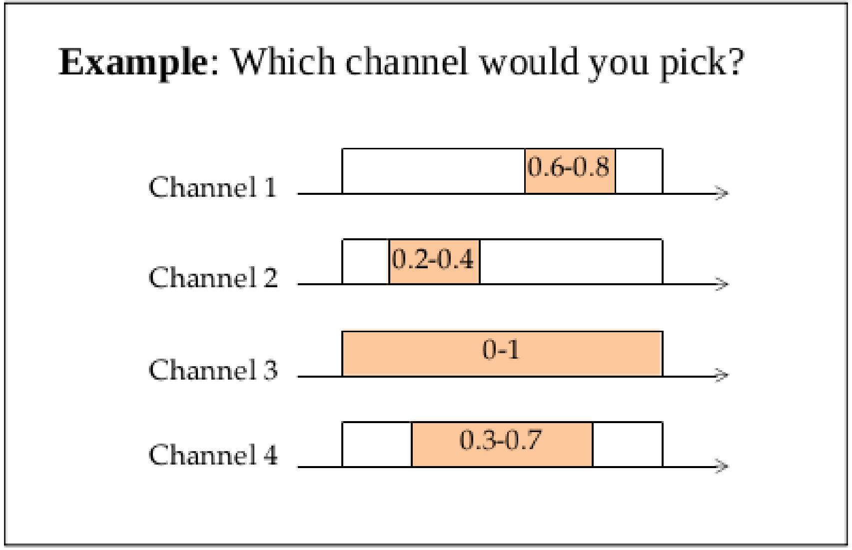


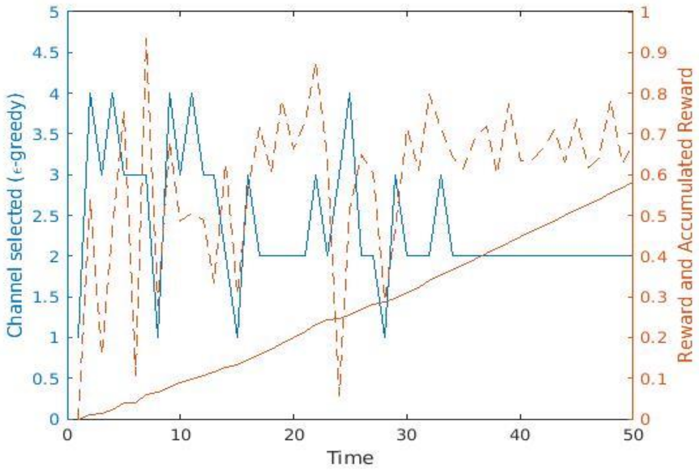


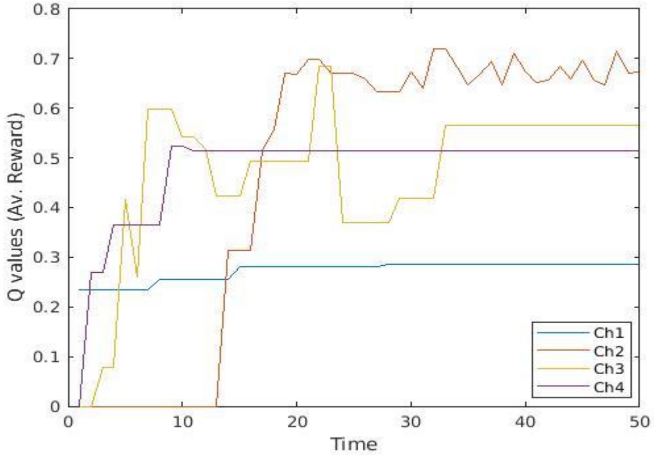


ε-greedy

- N=4 channels
- 1 episode, 50 iterations
- α=0.5; ε_0=0.8 (reduces 0.02 per step)

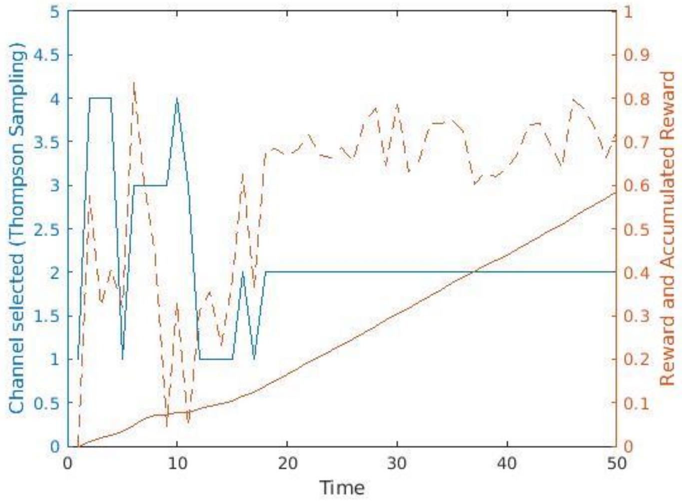






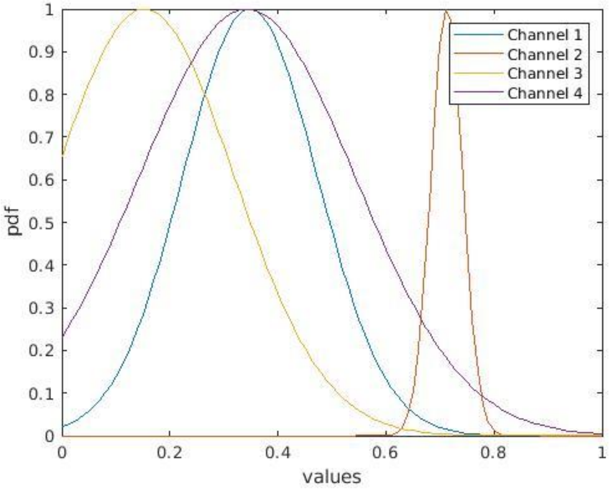
Thompson sampling

- For each action, we estimate the av. reward, and adjust the variance proportionally to the number of times we played that action in the past:
    - *the variance 'captures' how much confident we are on the future reward we will obtain when playing a certain arm.*
- TS includes the 'consolidated' knowledge of a certain action when deciding to take it or not: i.e., playing arm 1 may give me higher reward, but I have more uncertainty.



Algorithm 2: Multi-Armed Bandits Thompson Sampling

Initialize: $t = 1$, for each arm $a_k \in \mathcal{A}$, set $r_{k,0} = 0$ and $n_k = 0$
while *active* **do**
  For each arm $a_k \in \mathcal{A}$, sample $\theta_k(t)$ from normal distribution $\mathcal{N}(r_{k,t-1}, \frac{1}{n_k+1})$
  Play arm $a_k = \underset{k=1,\ldots,K}{\text{argmax}}\ \theta_k(t)$
  Observe the channel occupancy $\rho_{k,t}$
  Compute the reward $r_{k,t} = (1-\alpha)r_{k,t-1} + \alpha(1-\rho_{k,t})$
  $n_k \leftarrow n_k + 1$
  $t \leftarrow t + 1$
end



UCB - Upper Confidence Bound

- Brute force + pick the best ‘action’
- Optimistic Function to explore:
    - *if I have not tried an action for a while (as it was not good), perhaps now it has changed*
- The value of ‘c’ depends on the reward range of values, and allows to adjust the exploration

Algorithm 3: Multi-Armed Bandits UCB

Input: $\mathcal{A}$: set of possible actions $\{a_1, ..., a_K\}$

Initialize: $t = 1$, play each arm $a_k \in \mathcal{A}$ once

while *active* **do**

  Draw $a_k = \underset{k=1,...,K}{\operatorname{argmax}} \left(r_k + c\sqrt{\frac{2ln(t)}{n_k}} \right)$

  Observe the channel occupancy $\rho_{k,t}$

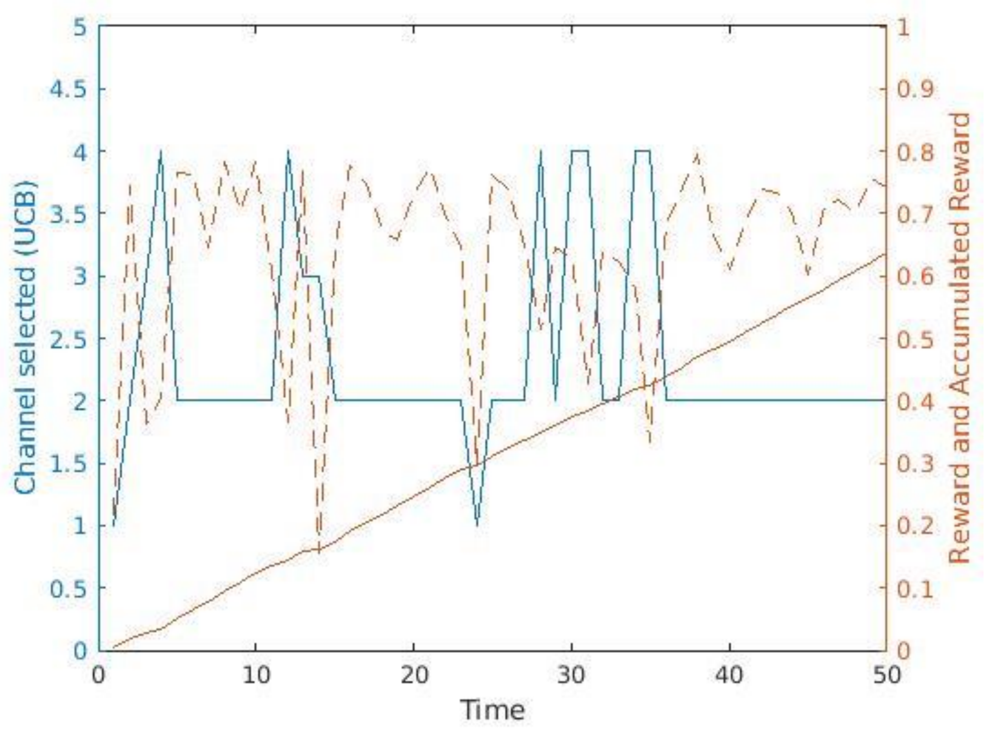
  Compute the reward $r_{k,t} = (1-\alpha) r_{k,t-1} + \alpha(1-\rho_{k,t})$

  $n_k \leftarrow n_k + 1$

  $t \leftarrow t + 1$

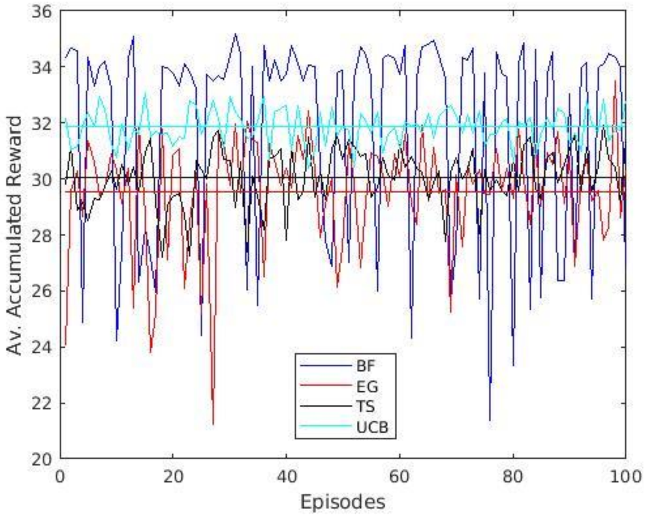
end

c=1/K

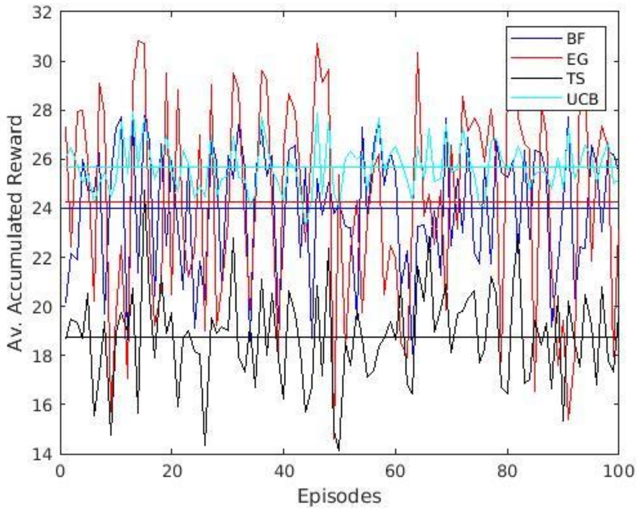


And the winner is…

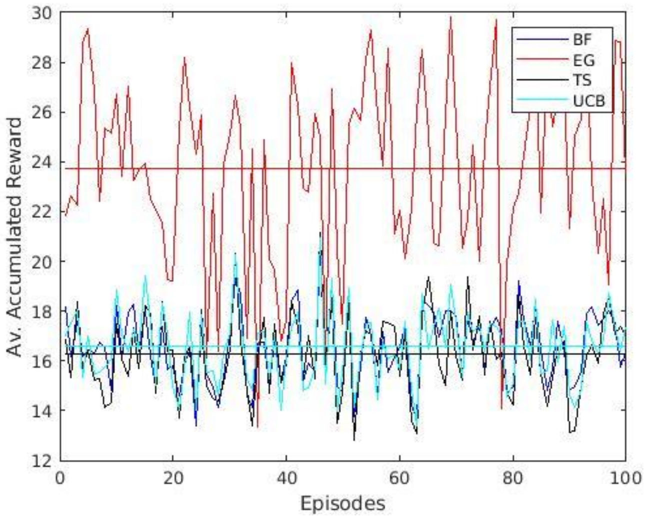
- We compare all 4 algorithms
- 100 episodes (each one of 50 iterations)
- (still 50 iterations per episode)



4 channels



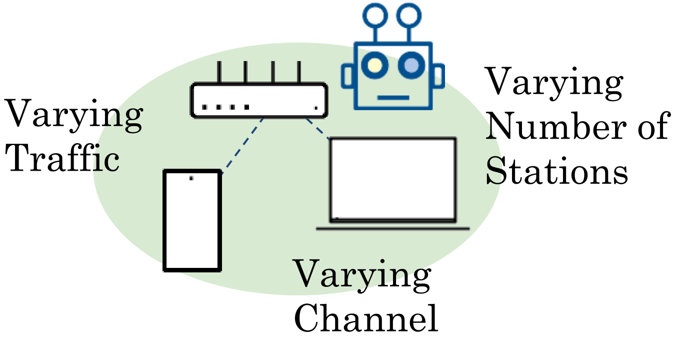
24 channels



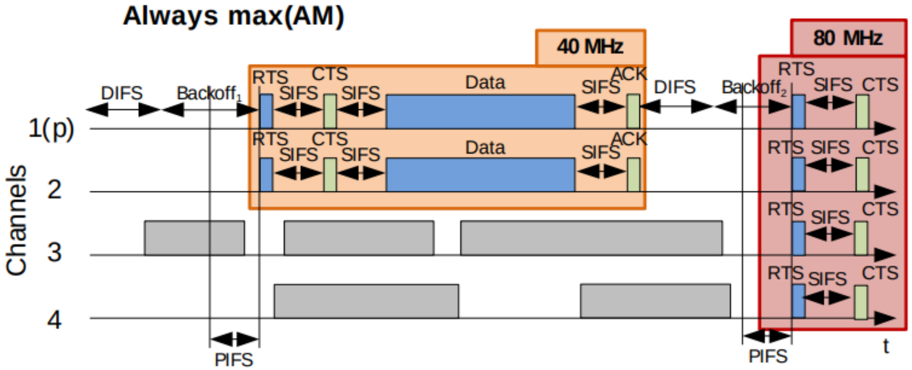
48 channels

Is it easy to implement?

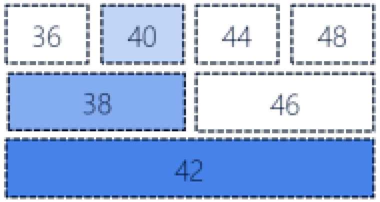
- The AP stores the estimated occupancy of every visited channel (including the actual one).
- The AP switches the Wi-Fi network periodically to a new channel following the ML algorithm using '<u>Channel Switch Announcement</u>' elements in beacons.
- Decision to switch (based on rewards and exploration rate) should take into account not only the channel occupancy but also network state and traffic load.
    - The AP can activate the 'channel search' agent only when current performance is below its needs.
- Inactivity periods can be used to gather statistics from the other channels if the system is stationary.



ML-enhanced Decentralized Spectrum Allocation

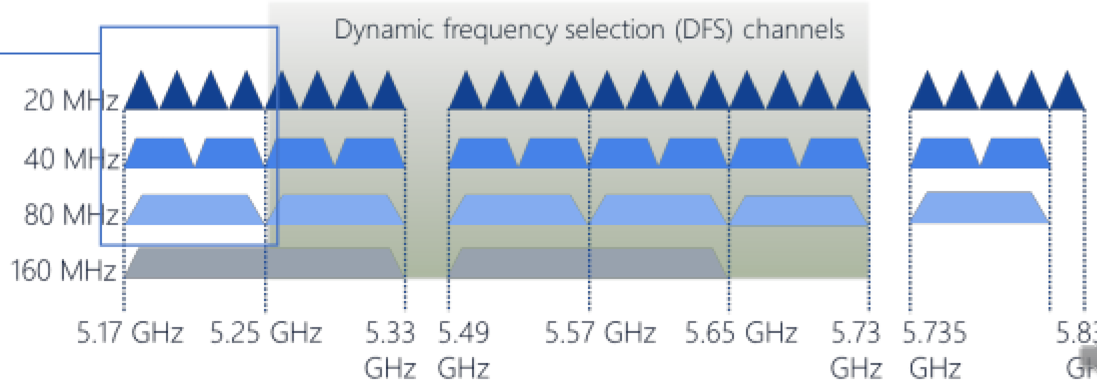


S. Barrachina-Muñoz, et al.. "[Multi-armed bandits for spectrum allocation in multi-agent channel bonding WLANs]()." IEEE Access 9 (2021): 133472-133490.



Primary 20 MHz: #40
Secondary 20 MHz: #36
Primary 40 MHz: #38
Secondary 40 MHz: #46

21

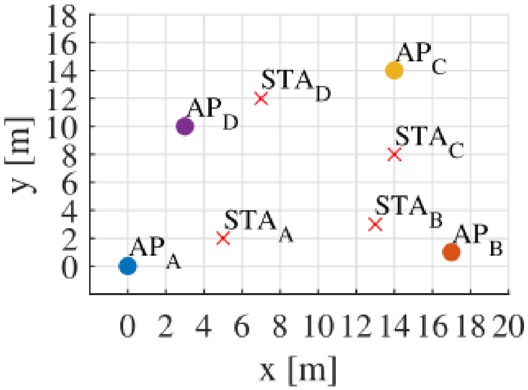


ML-enhanced Decentralized Spectrum Allocation

- “Realistic” Wi-Fi simulation environment ([Komondor]())
  - Path-loss, SINR, Dyn. Bandwidth Access, etc.
- Traffic load AP (only DL traffic): 50 Mbps
- 80 MHz available (shared by all 4 BSSs)
- W=20, 40 and 80 MHz

- 12 different channels (width, primary) → 12 actions per BSS
  - 4 x 20 MHz channels (1 position for the primary)
  - 2 x 40 MHz channels (2 positions for the primary)
  - 1 x 80 MHz channels (4 positions for the primary)

The goal is to ‘learn’ a good channel allocation as fast as possible



(a) Location of nodes.

	A	B	C	D
A	-	40	20	80
B	40	-	80	40
C	20	80	-	80
D	80	40	80	-

(b) Interference matrix.

ML-enhanced Decentralized Spectrum Allocation

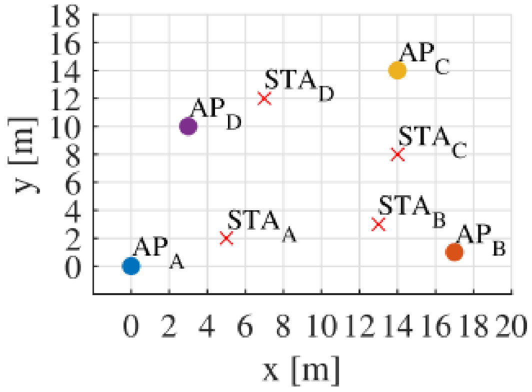
- Multi-armed Bandits (expfirst, e-greedy, TS, …)
- Available actions correspond to the set of available channels
    - 12 actions
- Rewards are computed taking into account the actual throughput during the simulation (in intervals of 5 s)

$$R(\xi) = \xi \qquad \xi_{w,t} = \frac{\Gamma_{w,t}}{\ell_{w,t}}$$

- 50 Mbps constant load (in average, Poisson traffic) per BSS
- Channel switching time not considered (few ms)

Practical considerations:

- Easy to implement on top of current 802.11 devices
- Channel switching overheads need to be considered



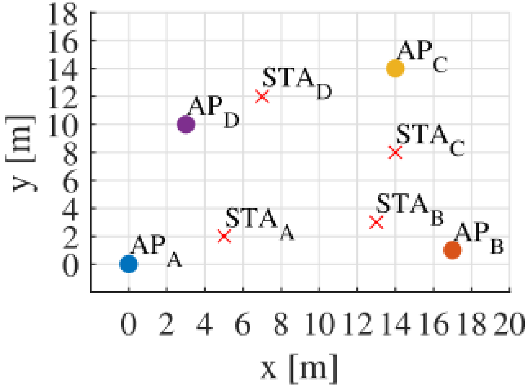
(a) Location of nodes.

	A	B	C	D
A	-	40	20	80
B	40	-	80	40
C	20	80	-	80
D	80	40	80	-

(b) Interference matrix.

ML-enhanced Decentralized Spectrum Allocation

- expfirst:
  - sequential exploration + actions sorted by reward (and stored)
  - action change when current av. reward decreases below others
- e-greedy, TS, UCB, EXP3



(a) Location of nodes.

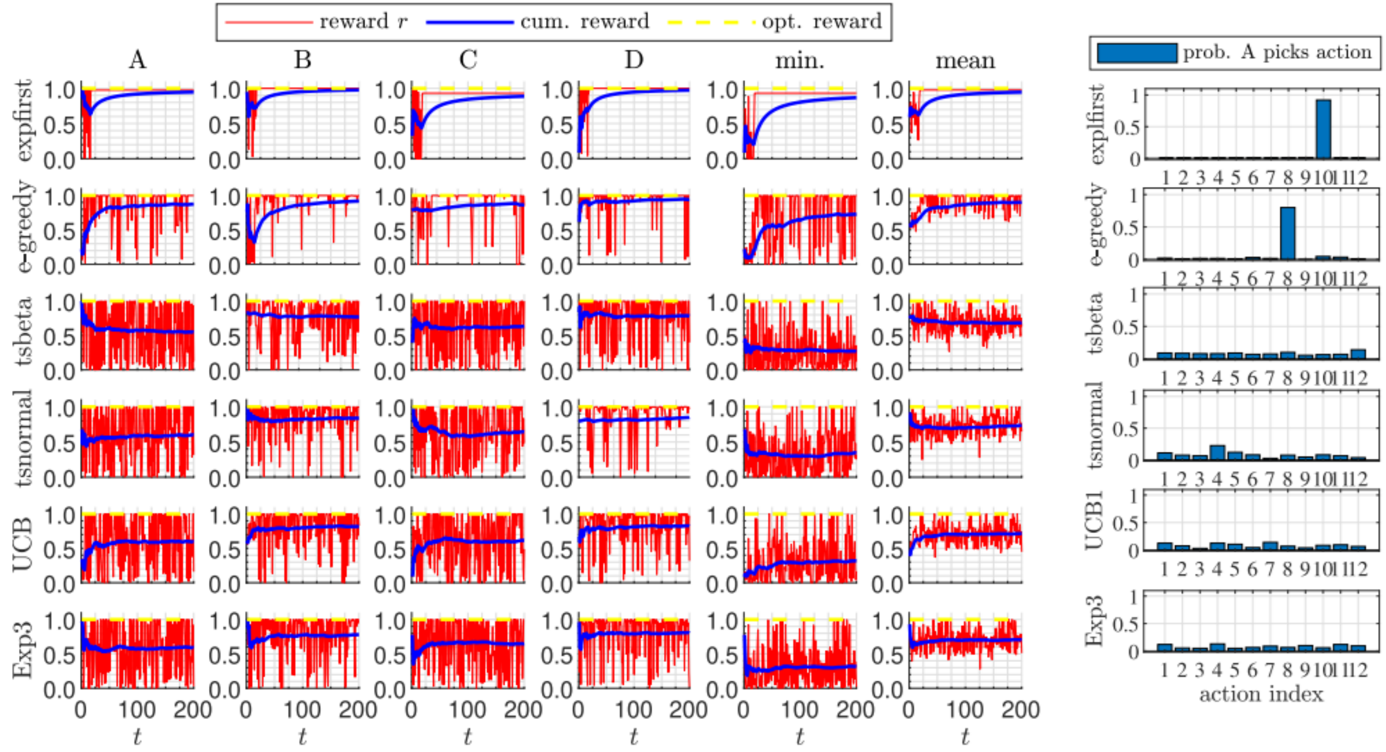
	A	B	C	D
A	-	40	20	80
B	40	-	80	40
C	20	80	-	80
D	80	40	80	-

(b) Interference matrix.

The data set:

https://zenodo.org/record/4265898#.Y5nS3NLMJ8I

ML-enhanced Decentralized Spectrum Allocation

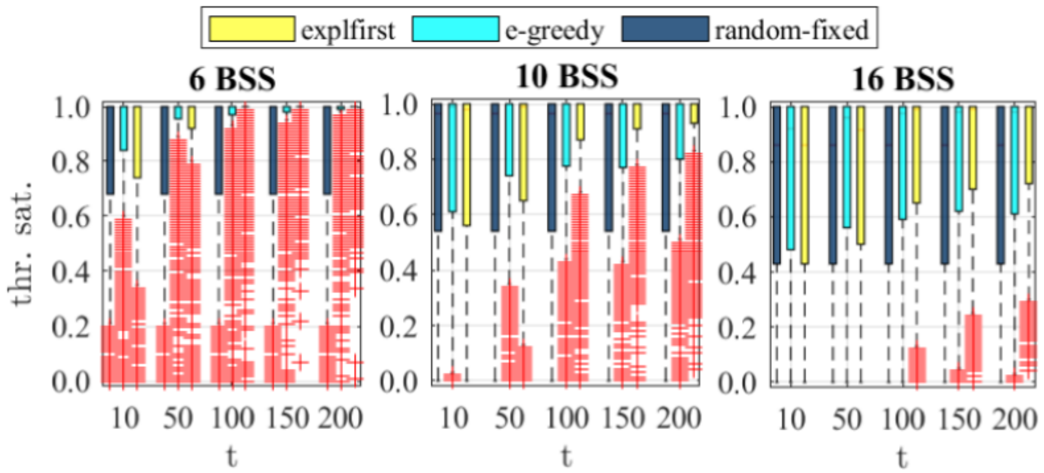


Unstable!
Why?

Only the simplest MAB algorithms perform well → they are able to find an acceptable equilibrium

ML-enhanced Decentralized Channel Selection

- Large deployments:
  - 20x20 m^2
  - 6, 10 and 16 BSSs
- Random per BSS traffic loads: 50, 100, 150 Mbps
- W = 20, 40, 80, 160 MHz
- 160 MHz available spectrum
  - 32 combinations of channels



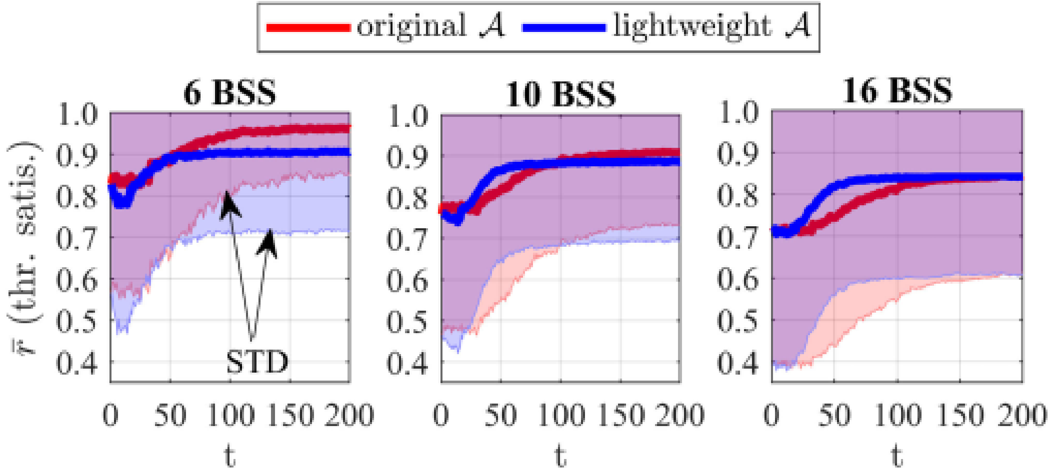
Both exploration-first and e-greedy can learn even in high-density deployments, outperforming by far static configurations. Besides, while exploration-first performs worse than e-greedy at the early stages of the simulation, it reaches much better performance as the simulation progresses.

Lightweighting The Action Space

- Is it possible to increase the learning convergence speed and reach good performance levels sooner?
- To that aim, we propose lightening the action space of each agent by reducing the possible values the maximum bandwidth attribute may take: from b ∈ {1, 2, 4, 8} to b ∈ {1, 8}, i.e., to allow only single-channel (b = 1) or to remove any bandwidth restriction (b = 8).
- Reduced action space (16 actions instead of 32)

So, is there any rule of thumb for reducing action spaces?

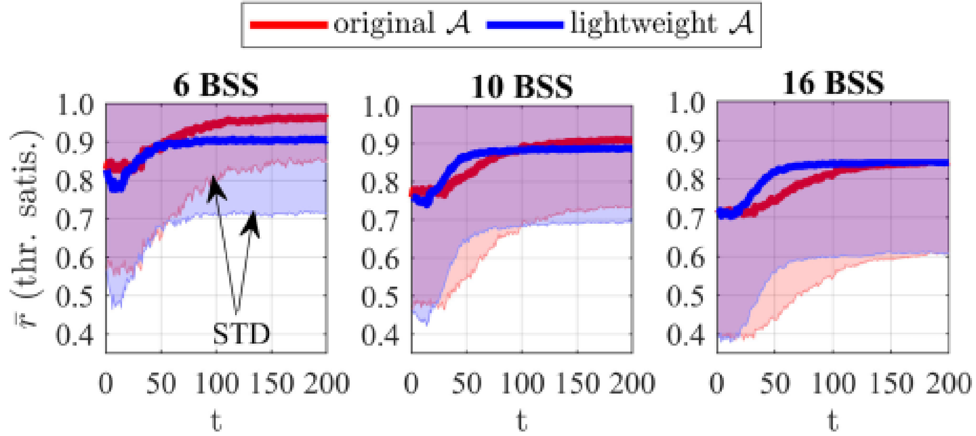
Adding new actions progressively?



Different Rewards

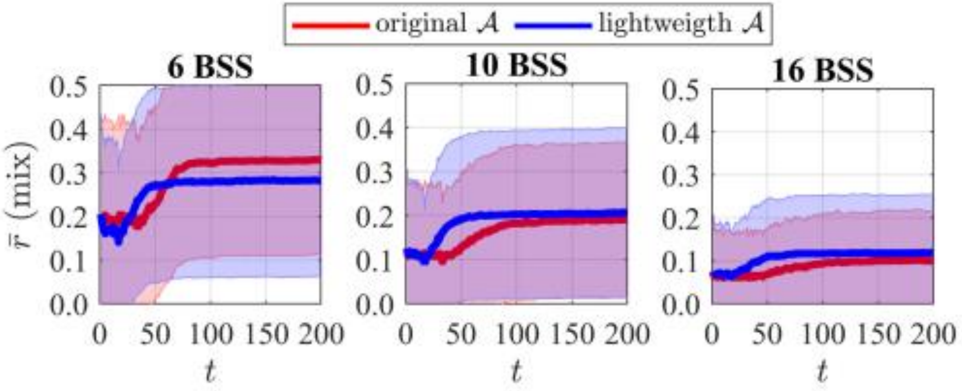
- Only throughput satisfaction:

$$R(\xi) = \xi \qquad \xi_{w,t} = \frac{\Gamma_{w,t}}{\ell_{w,t}}$$



- Throughput and delay:
    - Idea: *Given you get the required throughput, minimizing the delay means you are better using the available resources → more available resources for the other BSSs.*

$$R(\xi, \delta) = \xi \cdot \delta \qquad \delta_{w,t} = \frac{d^*}{d_{w,t}}$$

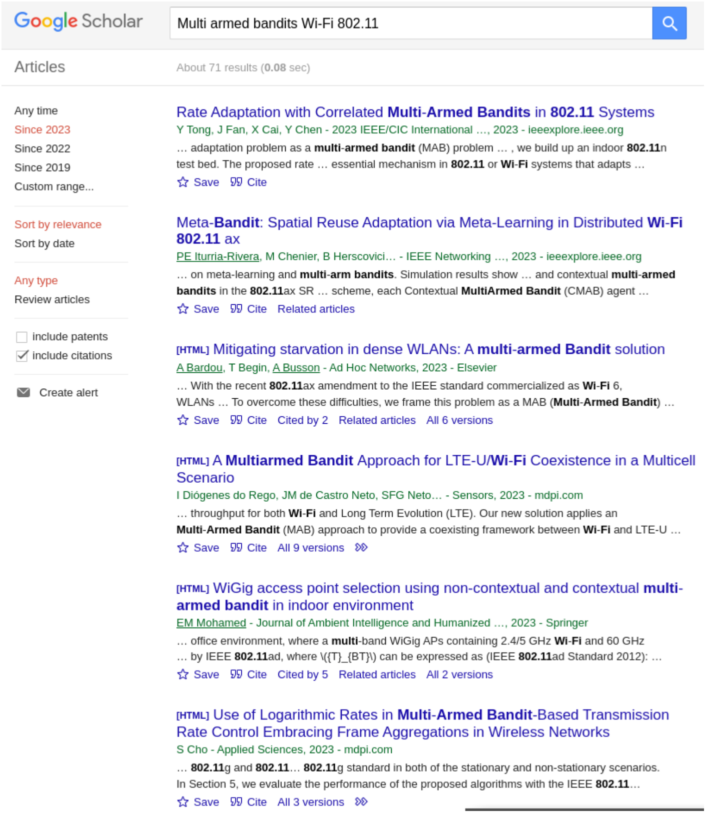


MAB Examples

- Channel selection
  - We introduce 3 basic MAB algorithms
  - S. Barrachina-Muñoz, et al.. "Multi-armed bandits for spectrum allocation in multi-agent channel bonding WLANs." IEEE Access 9 (2021): 133472-133490
- AP selection in multi-AP networks
  - A good lesson from multi-agent environments
  - M. Carrascosa, et al. "Multi-armed bandits for decentralized AP selection in enterprise WLANs." Computer Communications 159 (2020): 108-123.
- Reinforced-lib – universal RL library
  - Analysis of MABs for rate selection: ε-greedy, UCB, TS, EXP3, Softmax

Is the research community using MABs?

- Check our survey! Many papers using MABs.
- Some recent ones (Google Scholar)

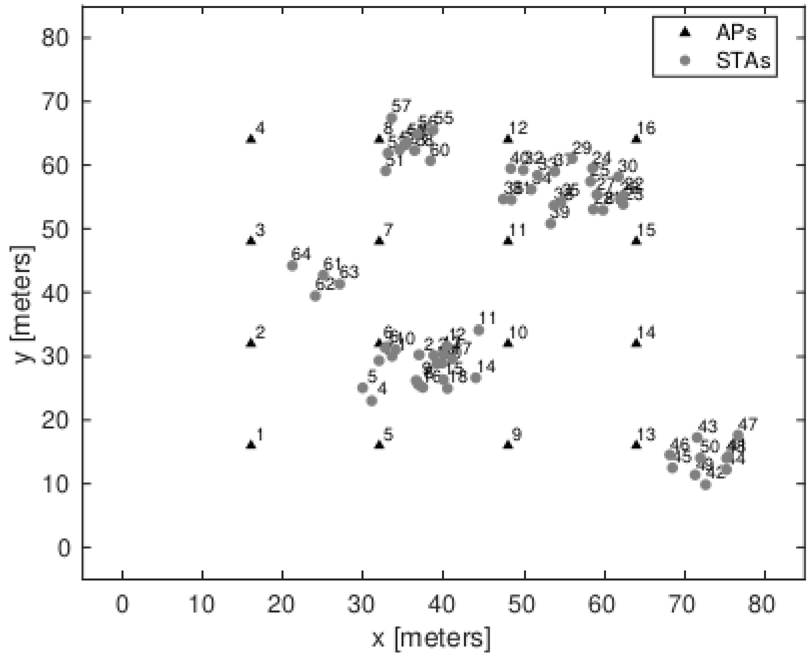


Since 2019: About 457 results (0.07 sec)

ML-enhanced Decentralized AP selection

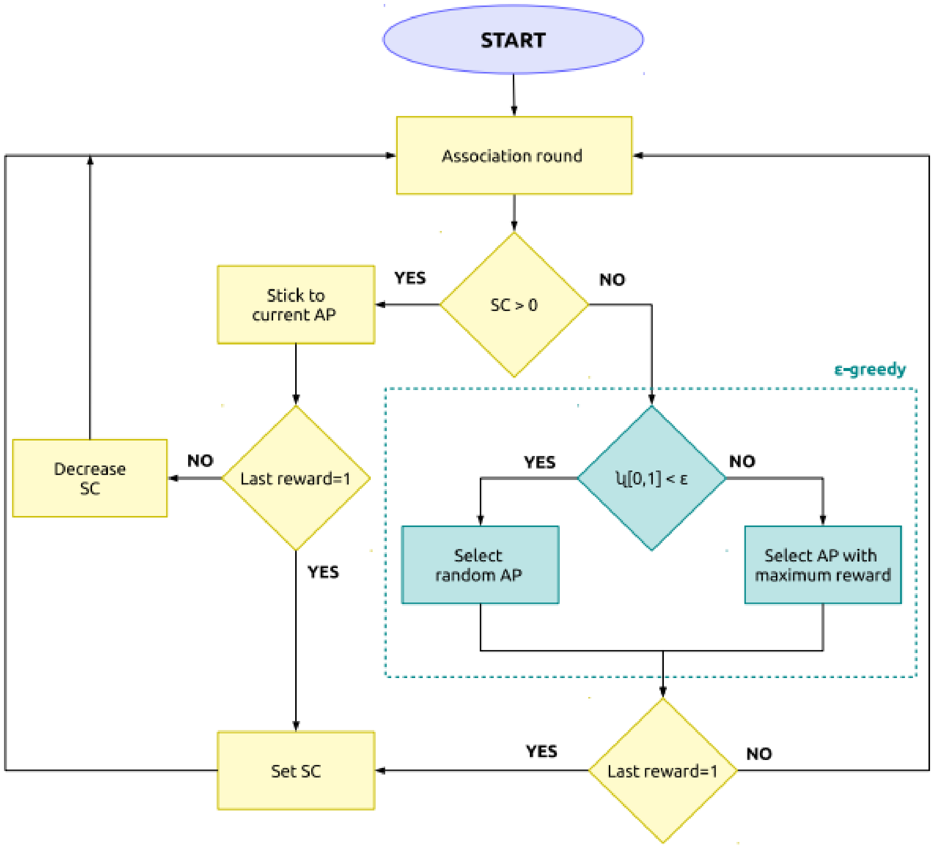
- Default AP selection is implemented considering only RSSI values
  - A station associates to the 'closest' AP
- This may cause that some APs are overloaded, while others are almost empty

- **<u>ML solution</u>**: STAs use an ML agent to find an AP that is good enough to satisfy their traffic requirements.
- The ML agent implements a MAB:
  - <u>Actions</u>: APs in sight
  - <u>Reward</u>: ratio Throughput/Load

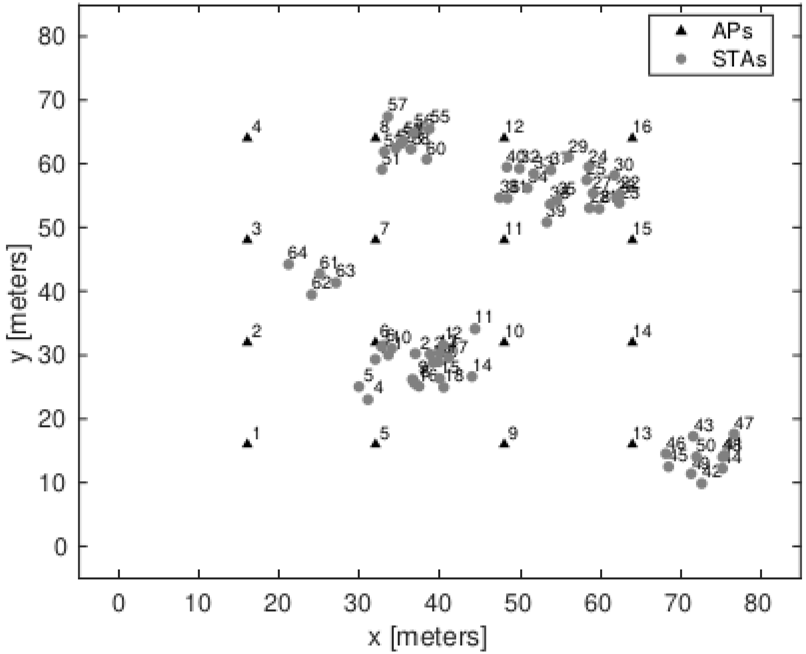


ML-enhanced Decentralized AP selection

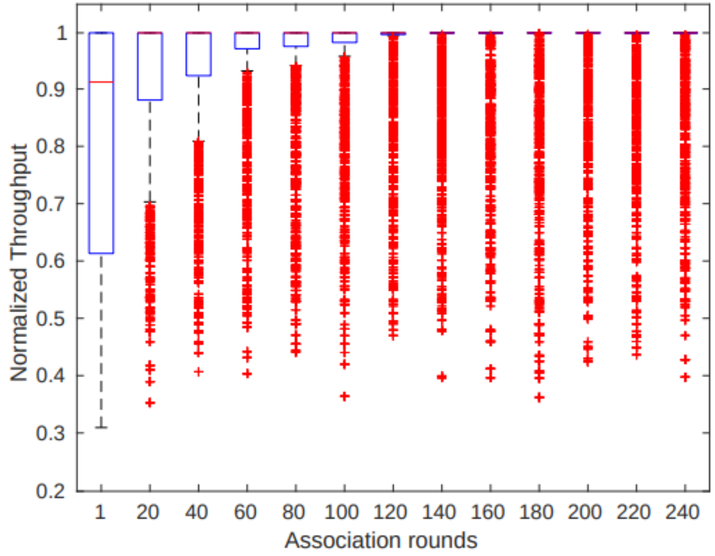
- Default ε-greedy MAB is enhanced with ‘**stickiness**’.
- **Stickiness** aims to improve stability by reducing the number of STAs switching APs simultaneously, or in a short time interval
- **Idea**: STAs stick to the last AP that satisfied their requirements for several association rounds before switching

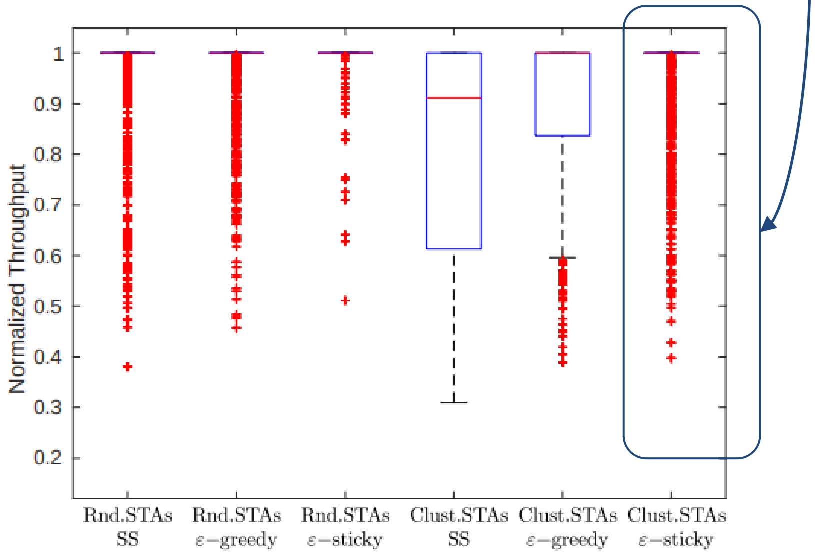


ML-enhanced Dec. AP sel.



- Non overlapping channel allocation
- 16 APs, 64 STAs
- Load STA: 4 Mbps





ML-enhanced Dec. AP sel.

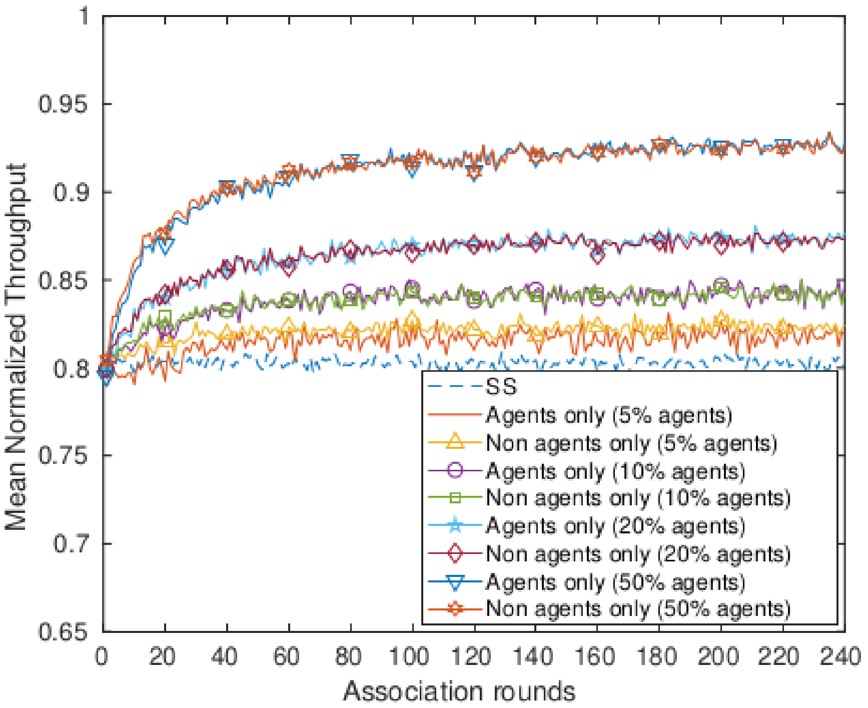


Figure 16: Mean normalized throughput of agent STAs vs. non agent STAs.

Pros:

- Easy to implement without any major update on 802.11 devices
- All network devices benefit even if they are not ML-enhanced
- Fully decentralized solution
- Reactive, adapts to any environment

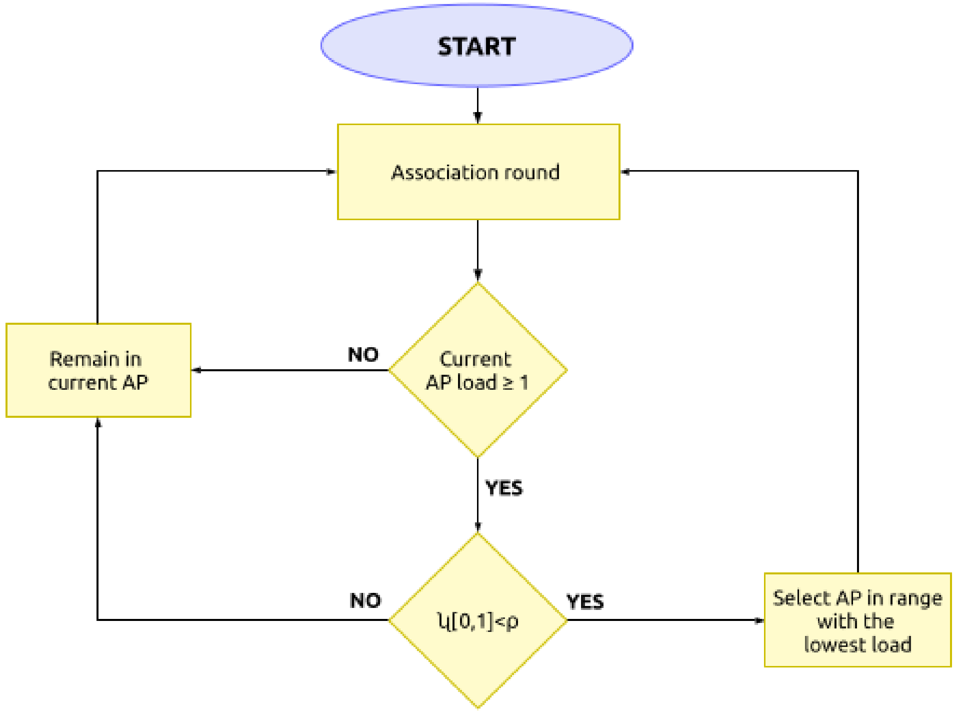
Cons:

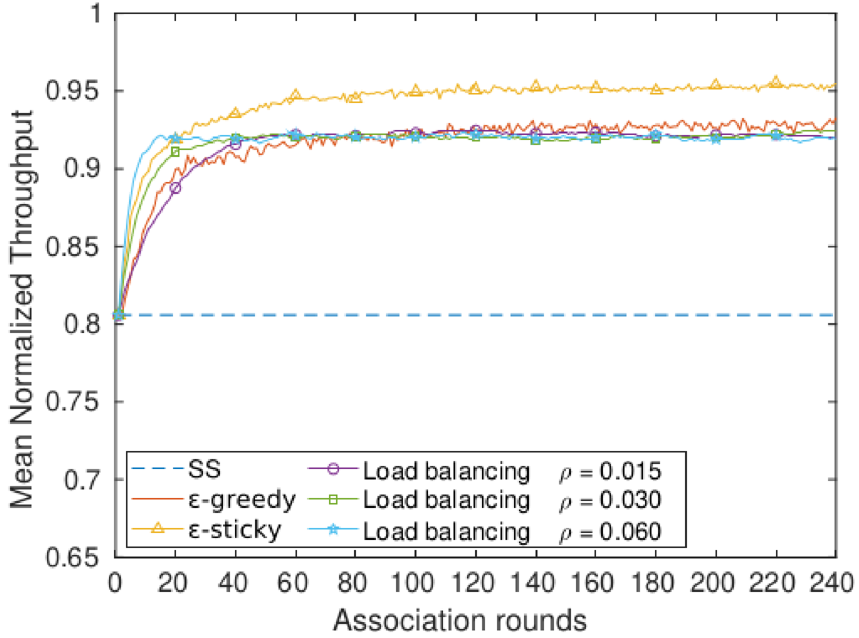
- Learning by interacting with the environment is required, implying periods of ‘low performance’ due to bad choices
- AP switching overheads

Practical aspects:

- Temporal window to assess throughput performance

ML-enhanced Dec. AP sel. vs Load-aware AP sel.



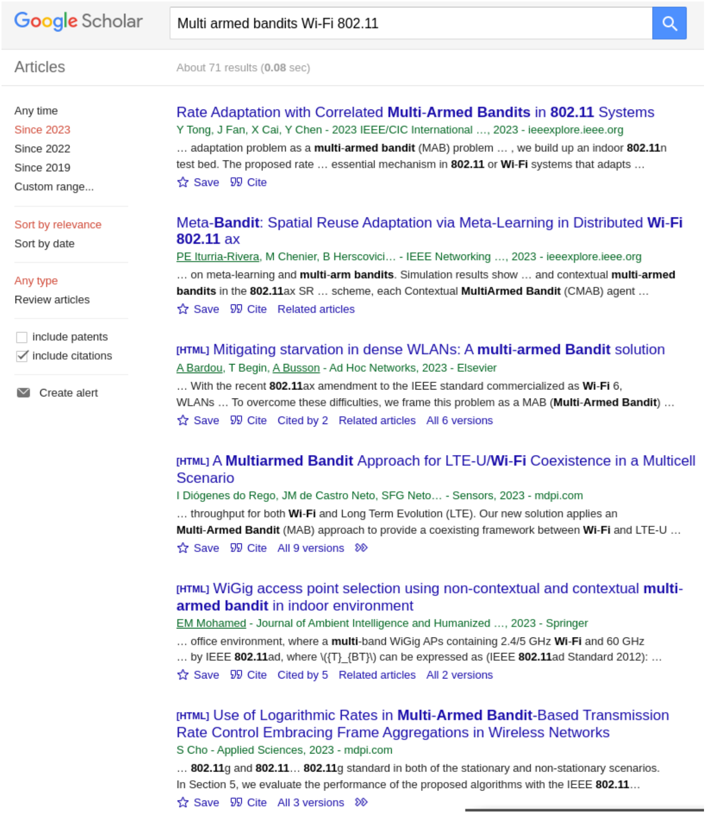


MAB Examples

- Channel selection
  - We introduce 3 basic MAB algorithms
  - S. Barrachina-Muñoz, et al.. "[Multi-armed bandits for spectrum allocation in multi-agent channel bonding WLANs]()." IEEE Access 9 (2021): 133472-133490
- AP selection in multi-AP networks
  - A good lesson from multi-agent environments
  - M. Carrascosa, et al. "[Multi-armed bandits for decentralized AP selection in enterprise WLANs]()." Computer Communications 159 (2020): 108-123.
- Reinforced-lib – universal RL library
  - Analysis of MABs for rate selection: ε-greedy, UCB, TS, EXP3, Softmax

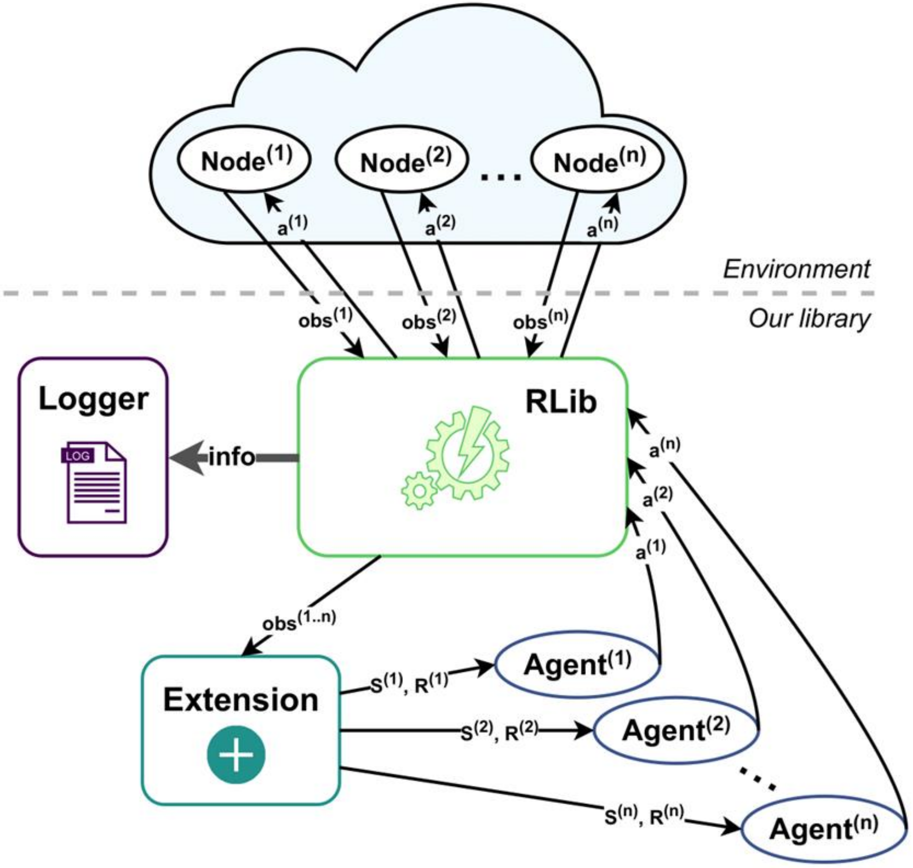
Is the research community using MABs?

- Check our survey! Many papers using MABs.
- Some recent ones (Google Scholar)



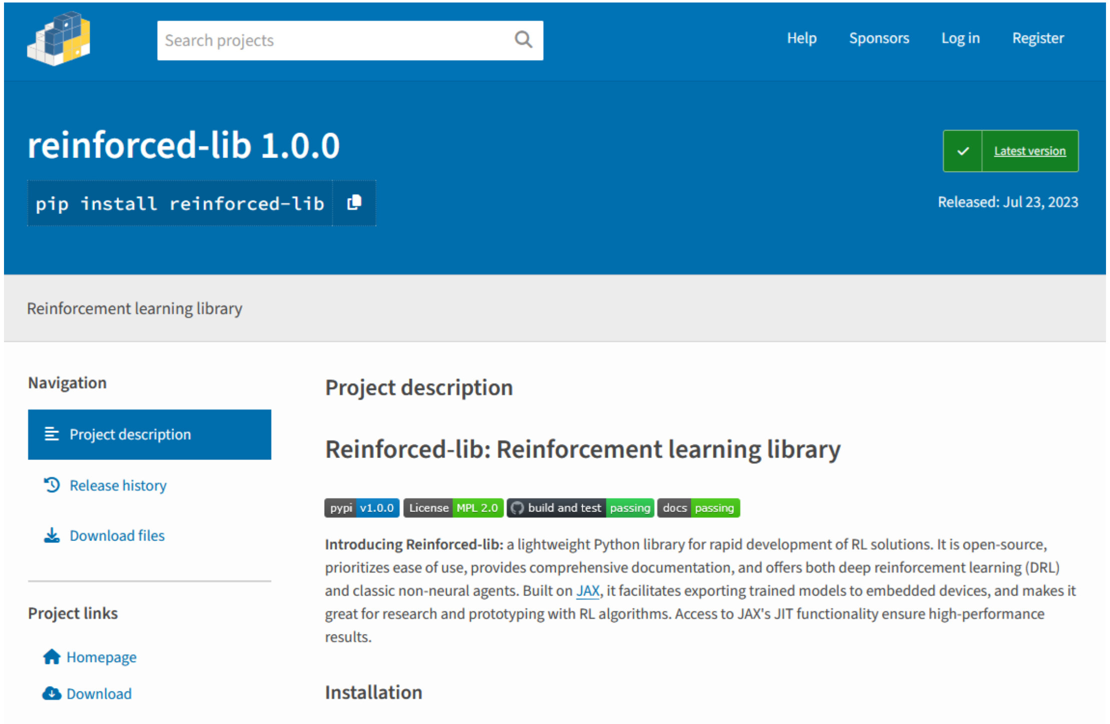
Since 2019: About 457 results (0.07 sec)

Reinforced-lib



https://github.com/m-wojnar/reinforced-lib

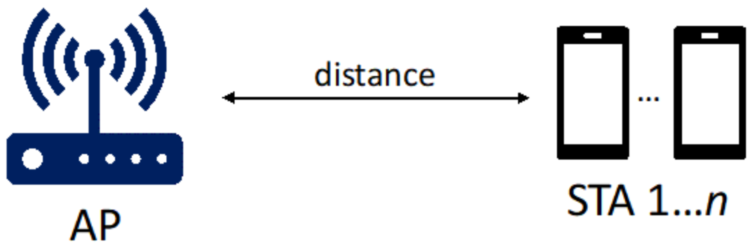
Reinforced-lib: Features

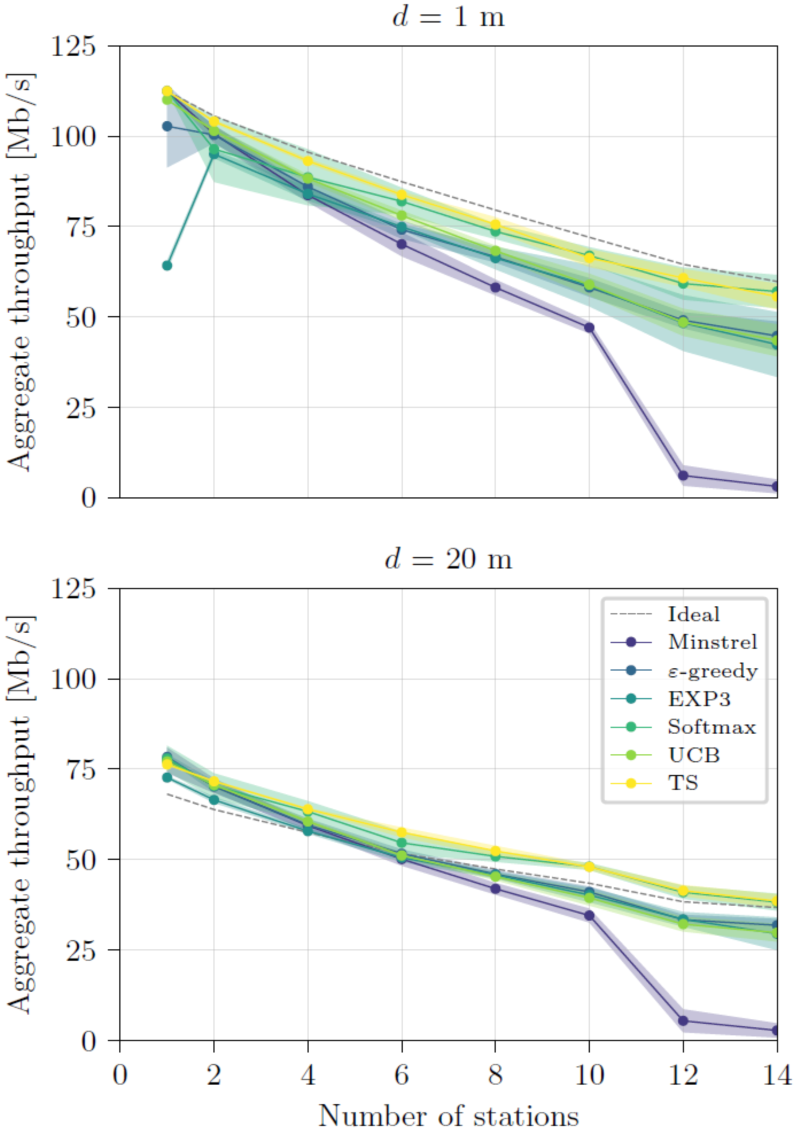


- Open-source
- Easy-to-use
- Extensive documentation
- Support for simple (MABs) and deep RL
- Export to embedded devices
- Provided examples
  - MAB rate selection
  - DQN CW selection

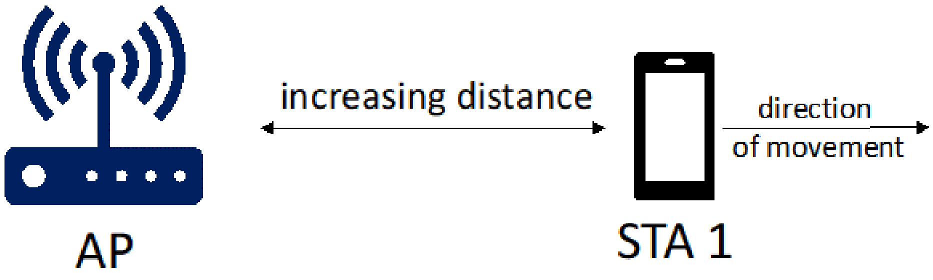
```
pip install reinforced-lib
```

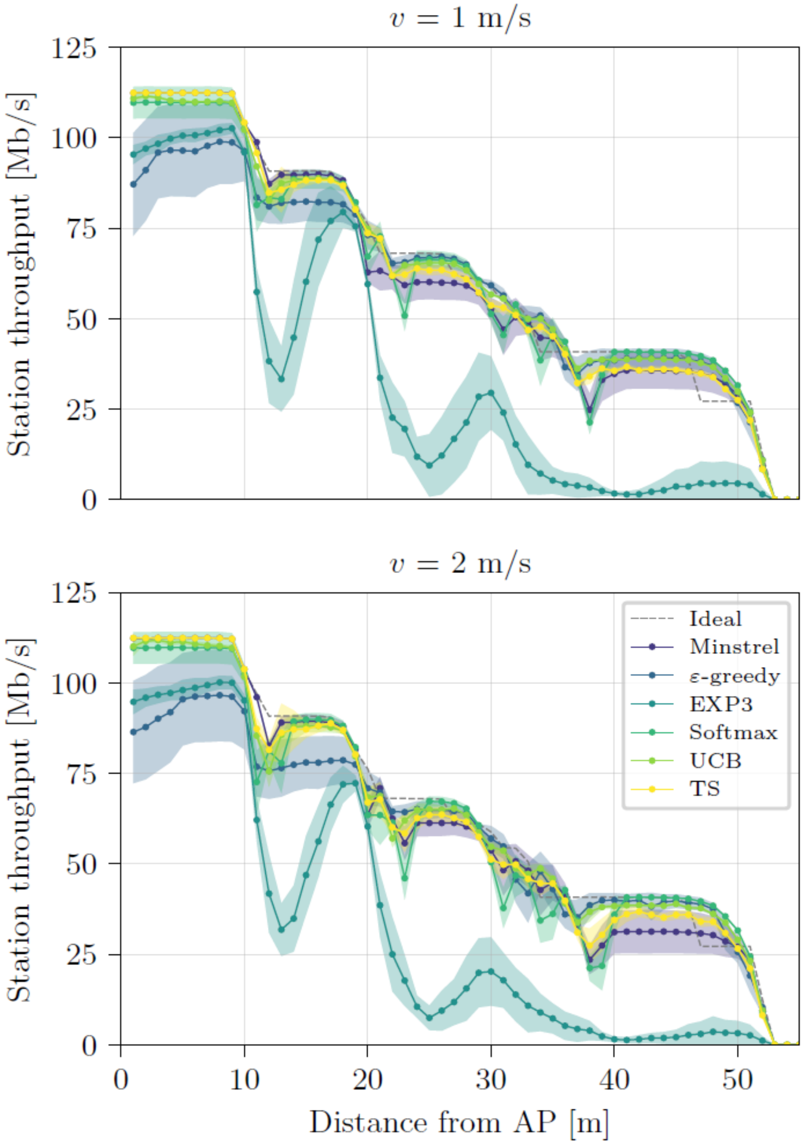
Reinforced-lib Example: Stationary Scenario





Reinforced-lib Example: Mobile Scenario





Re-cap

- MABs are simple, but powerful algorithms to make Wi-Fi networks ML-aware
  - They implement the exploration-exploitation trade-off
  - Engineering (parameter tuning) required to suit different problems
  - Convergence / responsiveness should be addressed / improved
- Examples:
  - Channel allocation and AP selection can be enhanced with MABs
  - In dynamic environments, lightweight approaches could be the best ones in terms of the r*esponse delay-performance* tradeoff
  - ML agents operation must contribute to reduce the environment randomness
  - Rate adaptation using MABs can improve traditional rate selection (e.g., Minstrel) algorithms
- Reinforced-lib: a tool for lightweight prototyping of RL solutions

Outline

Part 1: **Introduction. Why Wi-Fi may want to adopt AI/ML?** (30 mins) - Katarzyna

- Wi-Fi overview: A 30-year path from IEEE 802.11b to IEEE 802.11be, and beyond.
- Open challenges in Wi-Fi: Dealing with complexity and uncertainty.
- Requirements of next-gen WiFi networks.

Part 2: **A primer on AI/ML** (45 min) - Szymon & Boris

- Concepts, definitions, and overview of the main ML types (supervised, unsupervised, and reinforcement learning), including Wi-Fi examples.
- Popular ML techniques and paradigms: deep learning, reinforcement learning, online learning, federated learning.
- Deployment options: architecture, data handling, marketplaces.

Break (10:00-10:30)

Part 3: **Multi-Armed Bandits for Responsive Wi-Fi networks** (45 mins) - Boris & Szymon

- Multi-Armed Bandits: exploitation-exploration trade-off; e-greedy, Thompson Sampling, UCB.
- Examples: Channel Selection; AP selection;
- Reinforced-lib + Example (MCS selection).

Part 4: **Wi-Fi & ML: Practical considerations** (45 mins) - Francesc

- Wi-Fi & ML: A two-sided relationship
- Adoption of ML in Wi-Fi
- Hands-on Exercise II: Predicting the performance of Wi-Fi through Deep Learning

Part 5: **Open challenges, future research directions, and summary** (15 mins) - Katarzyna

- Open challenges and future research directions; Synergies with other disruptive technologies.
- Current AI/ML trends: LLM and Generative AI.
- Summary and takeaways.

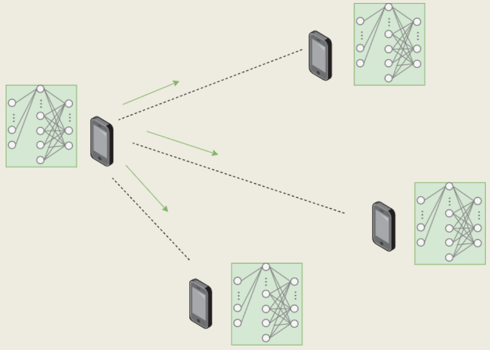
4 Wi-Fi & ML: Practical considerations

Wi-Fi & ML: a two-sided relationship
+
Adoption of ML in Wi-Fi
+
Hands-on Exercise II

Wi-Fi for ML & ML for Wi-Fi: A two-sided relationship

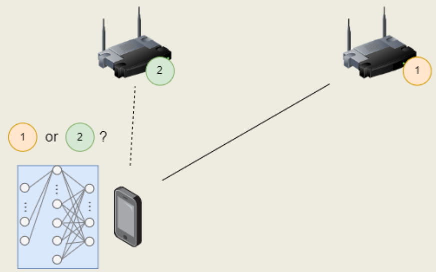
Wi-Fi for ML

- Rise of decentralization in ML (distributed/collective/federated intelligence)
- Real-time adaptation in ML applications
- Reach end-users, typically available wirelessly
- Wireless communications have become essential fuel for AI/ML
- Joint ML & communication optimization



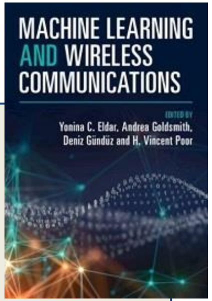
ML for Wi-Fi

- ML to address relevant problems in Wi-Fi
  - Network management & configuration
  - Advanced troubleshooting & anomaly detection
  - Network optimization (e.g., channel access, transmit power power adaptation, MCS selection optimization, etc.)
- On-top solutions vs Native AI/ML



Wireless for ML: Ongoing work & relevant directions

- AI/ML operations relying on wireless links
  - Exchange data/model parameters
  - Multi-agent coordination
  - Distributed training/inference
- Wireless resources have an impact on AI/ML methods
  - Provide timely results (e.g., spectrum resources)
  - Reliability, error-free transmissions (e.g., SINR)
  - Energy consumption (e.g., transmit power)



"Machine learning and wireless communications"

Yonina C. Eldar, Andrea Goldsmith, Deniz Gündüz, H. Vincent Poor

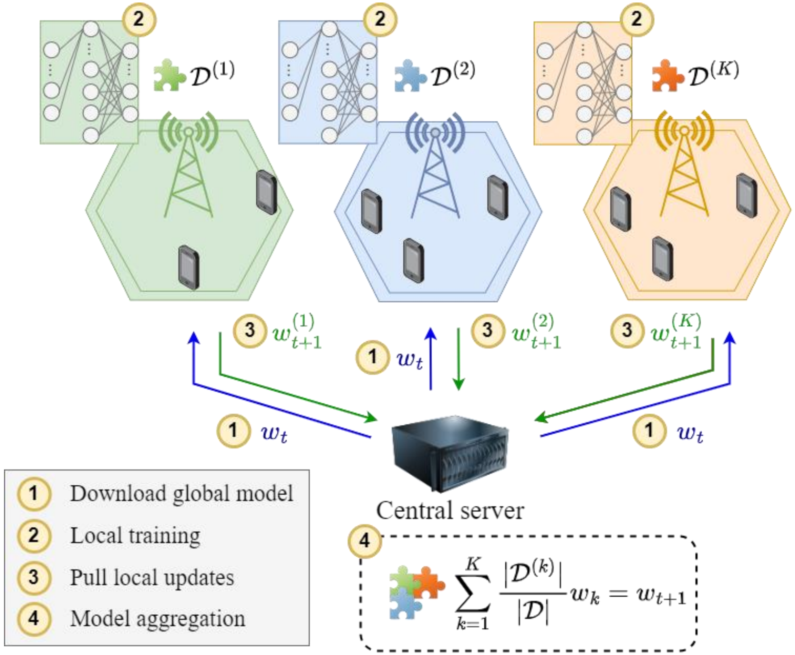
- Collaborative learning over WNs
- Optimized FL in WNs with constrained resources
- Quantized FL
- Over-the-air-computation
- Federated knowledge distillation
- Differential privacy in wireless FL
- Timely wireless edge interference

Examples

- Distributed learning over Wi-Fi:
  - Guerra, E., Wilhelmi, F., Miozzo, M. and Dini, P. (2023), "**The Cost of Training Machine Learning Models over Distributed Data Sources**," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2023.3274394.
- Over-the-air-computation
  - Amiri, M. M., & Gündüz, D. (2022). "**Over-the-Air Computation for Distributed Learning over Wireless Networks.**" Machine Learning and Wireless Communications, 434.

Distributed learning over Wi-Fi

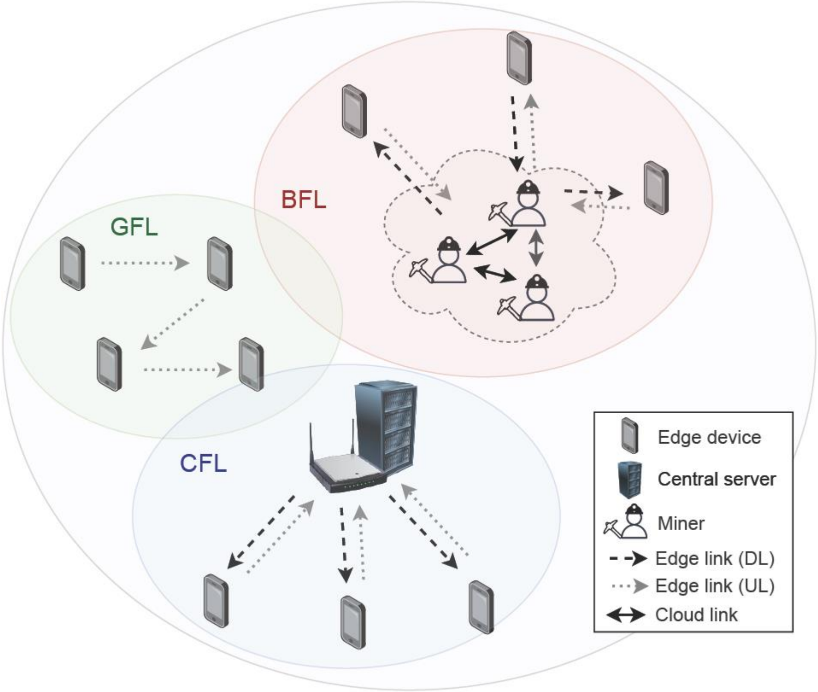
- Distributed ML trend
  - Improve efficiency of computation and communication resources
  - Bring AI optimization closer to users
  - Robustness, enhanced privacy, security...
- Federated Learning (FL)
  - FL devices exchange model weights rather than data
  - Communication efficiency
  - Improve scalability + Enhance privacy
- Strong dependency on (wireless) communication links
  - Significant cost of communication (overhead, energy)
  - Impact of communication latency into ML performance



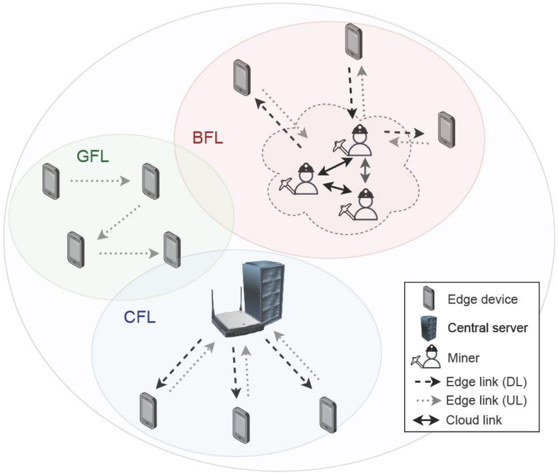
Distributed learning over Wi-Fi

Guerra, E., Wilhelmi, F., Miozzo, M. and Dini, P. (2023), "[The Cost of Training Machine Learning Models over Distributed Data Sources](#)," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2023.3274394.

- Three different FL settings
  a. **Centralized Federated Learning (CFL):** traditional setting with a central orchestrating server
  b. **Blockchained Federated Learning (BFL):** the central server is replaced by a distributed ledger
  c. **Gossip Federated Learning (GFL):** purely decentralized setting with P2P interactions
- Cost of training ML models
  a. Computation
  b. **Communication:**
     - Cloud link & **Edge link (UL/DL)**



Distributed learning over Wi-Fi



Guerra, E., Wilhelmi, F., Miozzo, M. and Dini, P. (2023), "[The Cost of Training Machine Learning Models over Distributed Data Sources](#)," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2023.3274394.

Algorithm	Time complexity	Communication Overhead
CFL	$O(RmE\|D_{\max}\|\|w\|)$	$2Rm\|w\|$
BFL	$O(R(\|w\|m^2 + E\|D_{\max}\|\|w\|m + 2^l + m\|w\|N_B))$	$R(\|w\|m^2 + \|w\|m + m\|w\|N_B)$
GFL	$O(RmE\|D_{\max}\|\|w\|)$	$Rm\|w\|$

Number of FL rounds

Model size (kB)

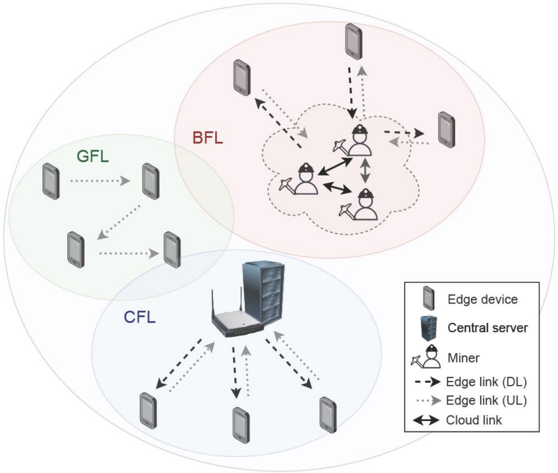
Number of clients/devices

Number of blockchain nodes

Size of popular ML models:

- FFNN = 800 KB
- CNN = 2.33 MB
- Resnet50 = 48 MB
- VGG19 = 79 MB

Distributed learning over Wi-Fi



Guerra, E., Wilhelmi, F., Miozzo, M. and Dini, P. (2023), "[The Cost of Training Machine Learning Models over Distributed Data Sources]," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2023.3274394.

Results of training a CNN in each setting for IID and non-IID (in parentheses) dataset distributions:

	Acc. Training	Acc. Validation	Acc. Test	Conv. Time (s)	Comp. Energy (%)	Tot. Energy (Wh)	Comm. overhead (GB)
CFL	0.99 (0.97)	0.97 (0.9)	0.96 (0.91)	125883.47 (124869.07)	99.65 (99.51)	201.68 (141.69)	186.4
BFL	0.99 (0.97)	0.97 (0.9)	0.96 (0.91)	131488.87 (130555.3)	99.95 (99.95)	1329.65 (1273.95)	37373.2
GFL	0.99 (0.67)	0.8 (0.22)	0.8 (0.22)	114217.66 (107854.86)	99.93 (99.84)	319.77 (143.22)	93.2

- Most of the energy spent is devoted to computation tasks (i.e., ML model training)
  - The communication energy is computed based on the power used for transmission
- **The communication overheads are significant**
  - **Need for a new class of traffic: AI/ML category (under discussion in standardardization)**

A significant portion of the literature analyzes ways to decrease the communication overheads generated by FL (compression, quantization, resource allocation, etc.)

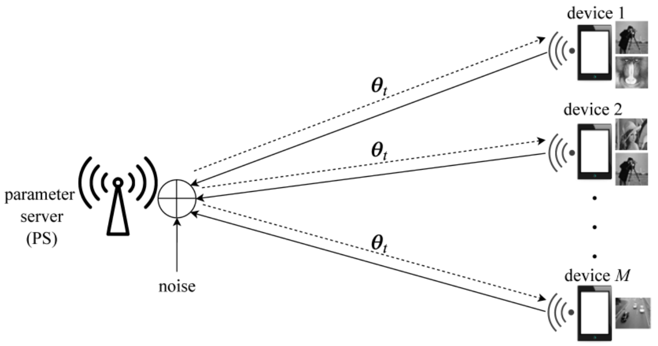
Yang, Z., Chen, M., Saad, W., Hong, C. S., & Shikh-Bahaei, M. (2020). Energy efficient federated learning over wireless communication networks. IEEE Transactions on Wireless Communications, 20(3), 1935-1949

Examples

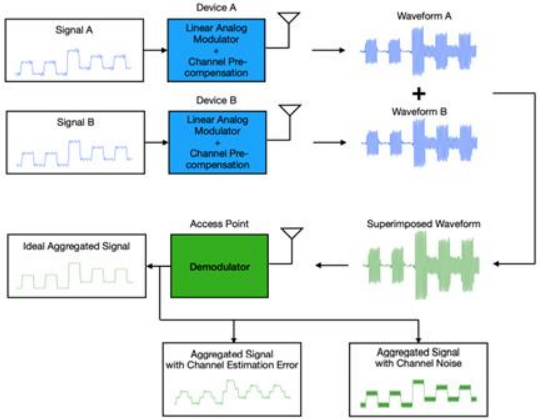
- Distributed learning over Wi-Fi:
  - Guerra, E., Wilhelmi, F., Miozzo, M. and Dini, P. (2023), "**The Cost of Training Machine Learning Models over Distributed Data Sources**," in IEEE Open Journal of the Communications Society, doi: 10.1109/OJCOMS.2023.3274394.
- Over-the-air-computation
  - Amiri, M. M., & Gündüz, D. (2022). “**Over-the-Air Computation for Distributed Learning over Wireless Networks.**” Machine Learning and Wireless Communications, 434.

Over-the-air-computation (OAC)

- OAC leverages interference from simultaneous transmissions for ML model aggregation/averaging
- The medium performs the computation (signal superposition)
- Reduce the number of computations
- The underlying communication infrastructure needs to be aware of the mechanism:
  - Schedule devices
  - Synchronization
  - Power control



Amiri, M. M., & Gündüz, D. (2020). Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air. IEEE Transactions on Signal Processing, 68, 2155-2169.

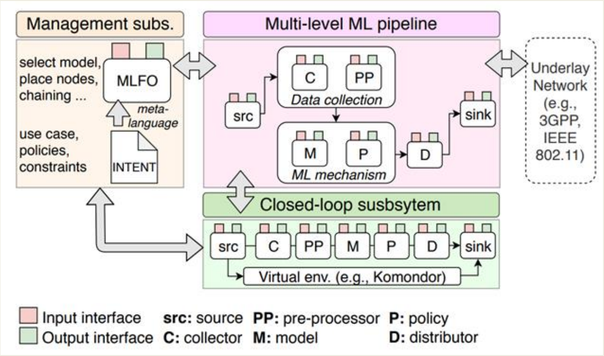


Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., & Cui, S. (2020). A joint learning and communications framework for federated learning over wireless networks. IEEE Transactions on Wireless Communications, 20(1), 269-283

Adoption of ML in Wi-Fi

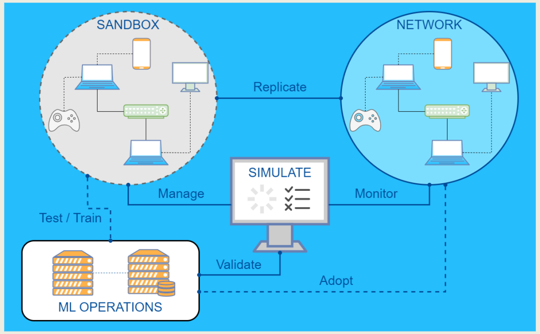
Architectural solutions

- Need for procedures, interfaces and data elements to support ML operations
- The ITU-T ML architectural framework
- Use case: AI/ML-based roaming enabled by the ITU-T's architecture



ML Sandbox / Digital twin

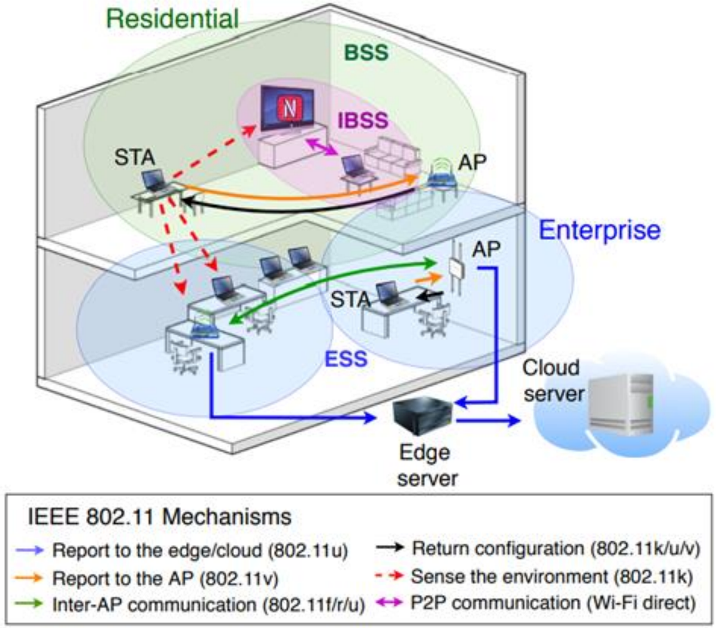
- Need of tools to trust AI/ML
- The ML Sandbox allows training, testing, and evaluating AI/ML solutions before being applied to production networks
- The role of network simulators as future network digital twins



Architectural integration of AIML in Wi-Fi

Challenges for ML adoption in Wi-Fi:

- IEEE 802.11 defines MAC/PHY only (limited AI functionalities with respect to 3GPP)
- Uncertainty due to the distributed channel access
- Different operation modes
  - Fully decentralized, with autonomous operation (e.g., residential)
  - Coordinated, allowing for groups of collaborating APs (e.g., Wi-Fi 8)
  - Centralized, operated from the cloud (e.g., industrial/enterprise Wi-Fi)
- Backward legacy compatibility (fair AI solutions are required)



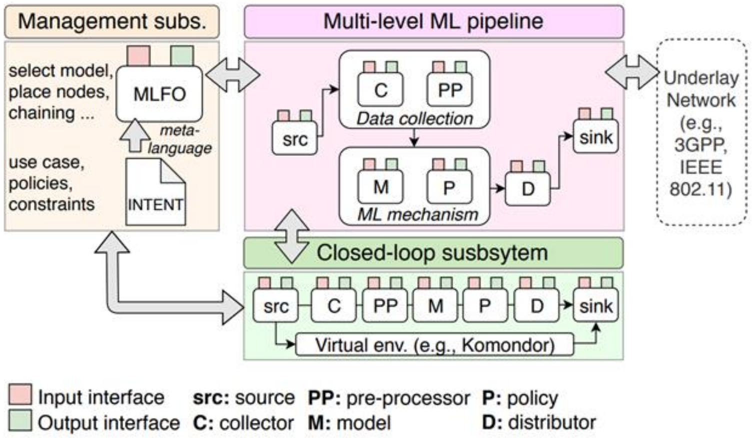
Existing AIML frameworks

ITU-T Y.3172	ITU's standardized architecture provides a common nomenclature for ML-related mechanisms so that interoperability with other networking systems is achieved
ITU-T Y.3174	Framework for data handling to enable machine learning in future networks including IMT-2020
ITU-T Y.3176	Machine learning marketplace integration in future networks including IMT-2020
ITU-T Y.3182	Architectural framework for Machine Learning Sandbox in future networks including IMT-2020
ETSI GS ZSM 002	ETSI/ZSM's architecture defines a service-centric architectural model to define at a high level a set of management services, including management, communication, and interoperation, for zero-touch network and service management
ETSI GS ZSM 008	Cross-domain E2E service lifecycle management
ETSI GS ZSM 009	Closed-Loop Automation
TR 37.817	3GPP's study on Artificial Intelligence and Machine Learning (AI/ML) in 5G RAN architectures
RP-220635	Work Item on Artificial Intelligence (AI)/Machine Learning (ML) for NG-RAN **[ONGOING – from March 2022]**

The ITU-T's Flexible ML Architecture

- **ML operations**
  - Multiple ML pipeline nodes to be instantiated for multiple purposes
    - Data collection, model inference, output distribution…
  - Specialized M&O is required (MLFO)
    - Chain ML pipeline elements
    - Deploy models, re-train, select data, etc.
- **ML sandbox (closed-loop subsystem)**
  - Safe environment to assist the ML operation
  - Increase trustworthiness on AIML
  - Speed-up model training/validation

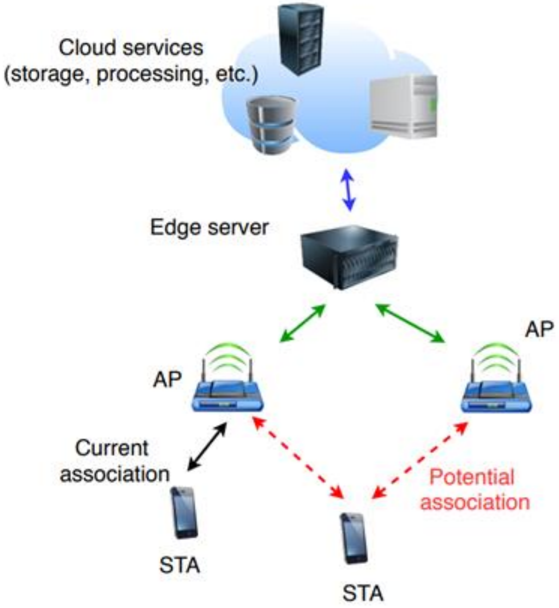
Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). **A flexible machine-learning-aware architecture for future WLANs**. IEEE Communications Magazine, 58(3), 25-31.

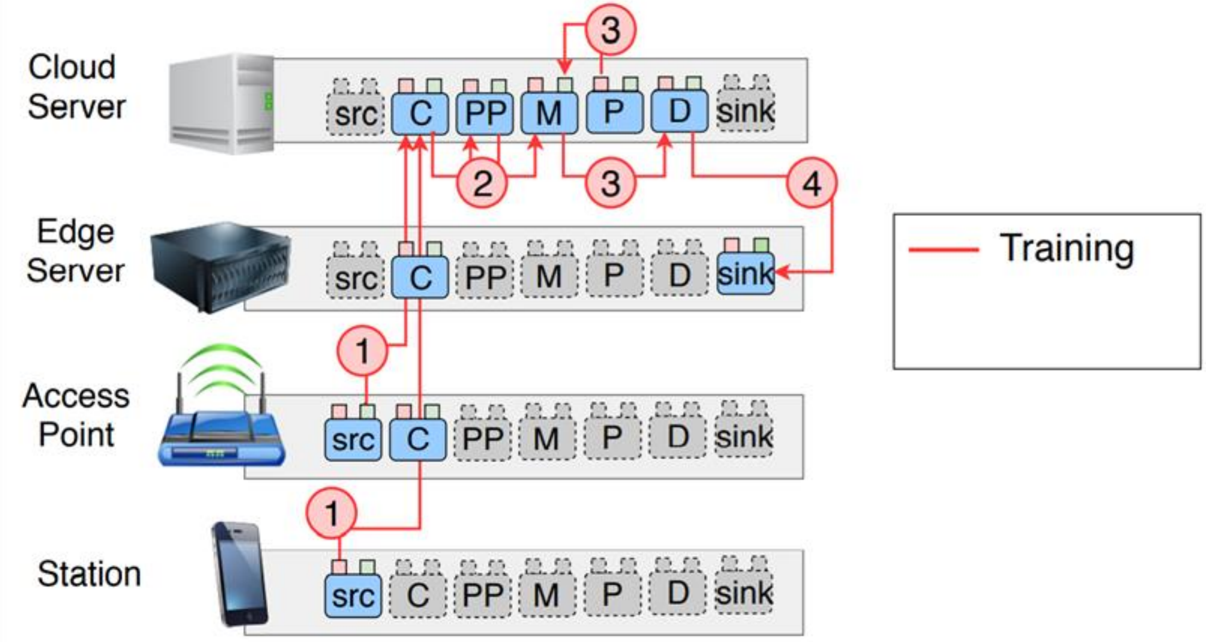


- Source (src): generates data to be used by the ML mechanism
- Collector (C): collects the data generated by sources
- Pre-processor (PP): prepares the data collected for its utilization by the ML mechanism
- ML model (M): applies the ML model specified by the intent
- Policy (P): provides a constraints/guidelines that delimit the behavior of the model
- Distributor (D): spreads the ML output across all the corresponding targets (or sinks)
- Sink: applies the ML output that is received from the distributor

ITU-T ML architecture's realization in Wi-Fi

AIML-based roaming

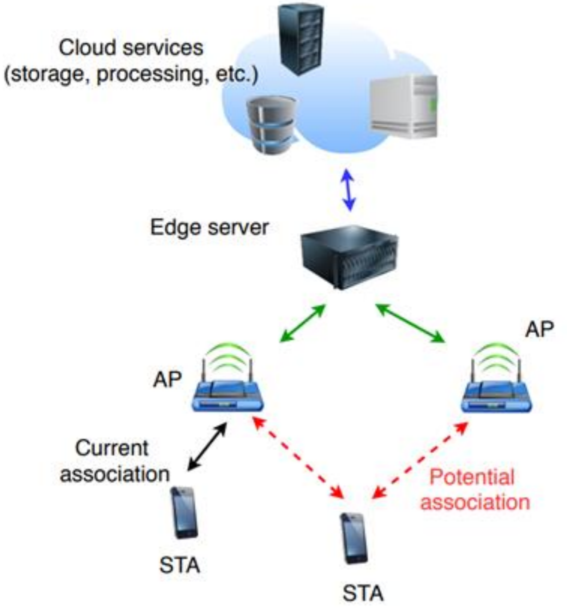


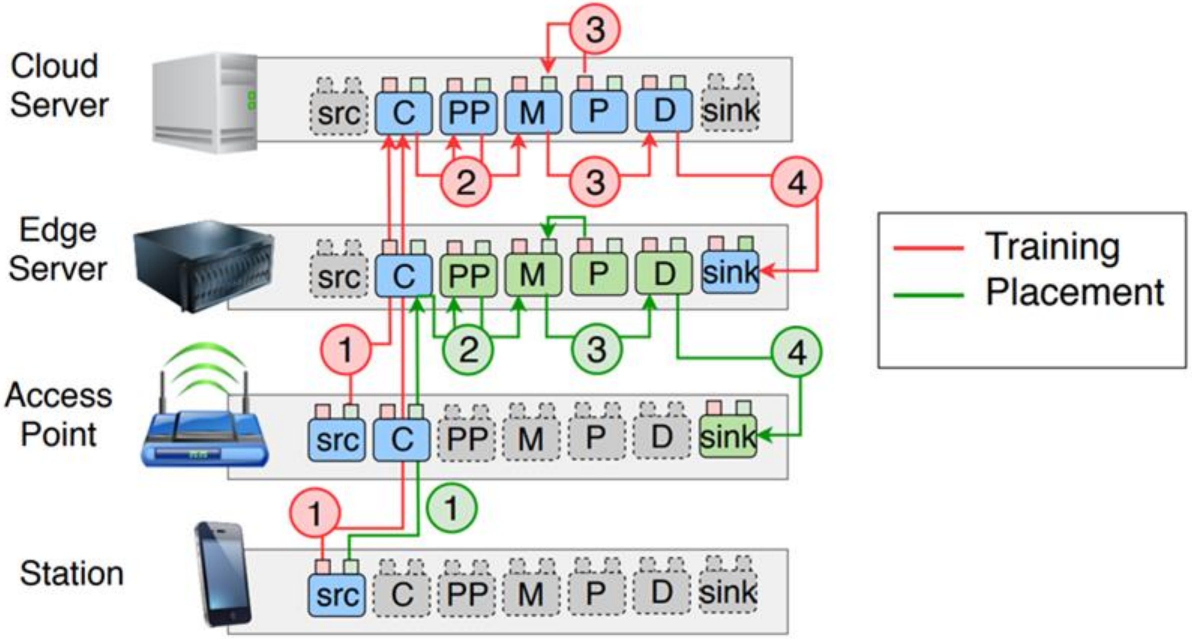


Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). [A flexible machine-learning-aware architecture for future WLANs](). IEEE Communications Magazine, 58(3), 25-31.

ITU-T ML architecture's realization in Wi-Fi

AIML-based roaming

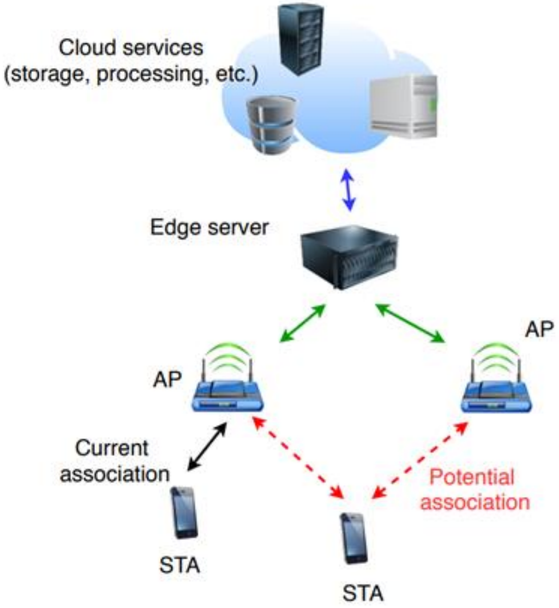


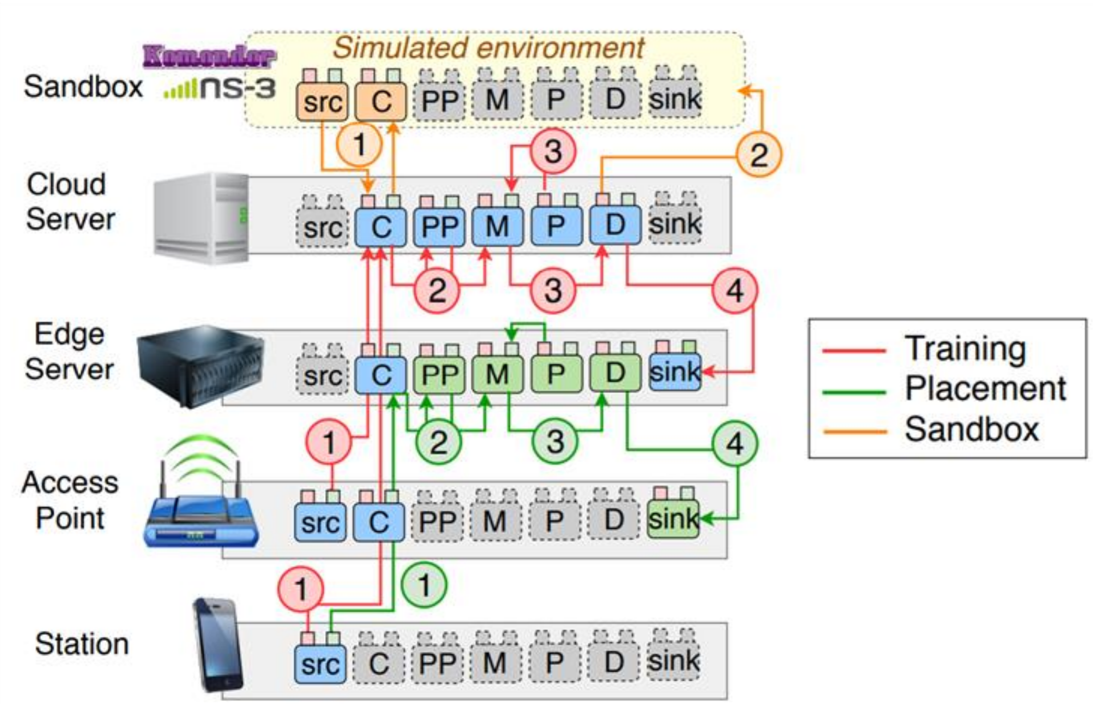


Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). [A flexible machine-learning-aware architecture for future WLANs](). IEEE Communications Magazine, 58(3), 25-31.

ITU-T ML architecture's realization in Wi-Fi

AIML-based roaming

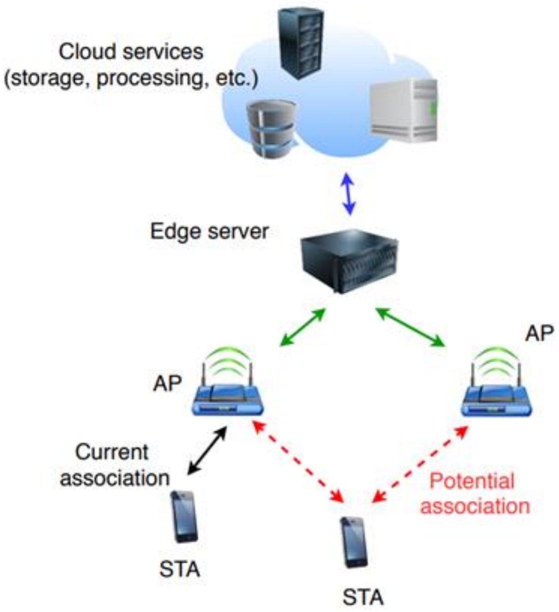


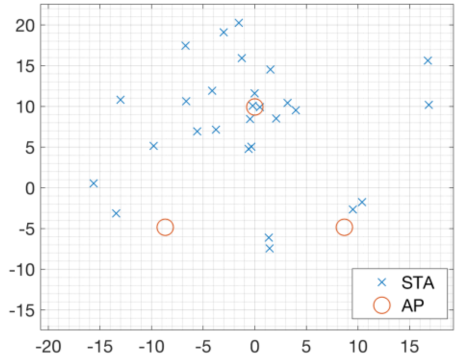


Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). [A flexible machine-learning-aware architecture for future WLANs](). IEEE Communications Magazine, 58(3), 25-31.

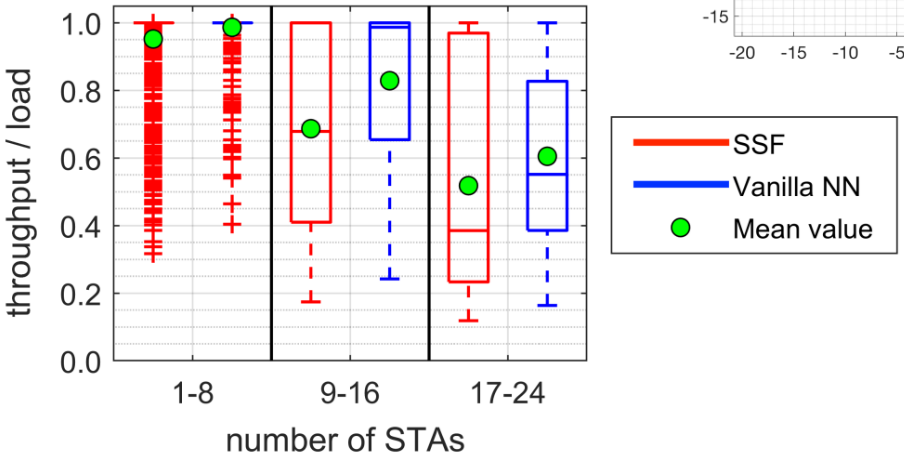
ITU-T ML architecture's realization in Wi-Fi

AIML-based roaming





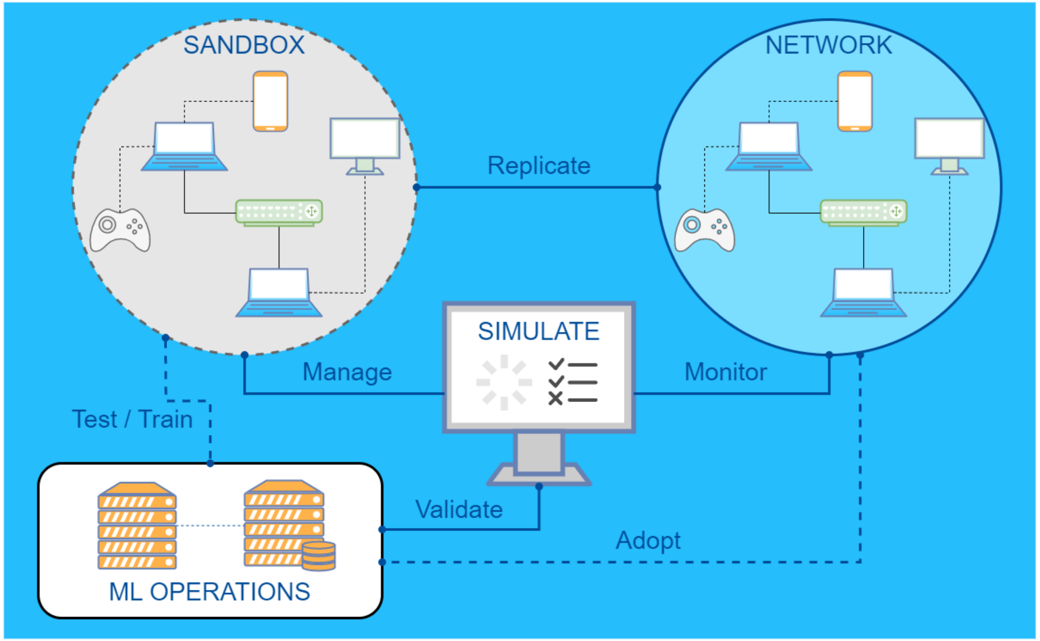
Strongest Signal First (SSF) vs Machine Learning

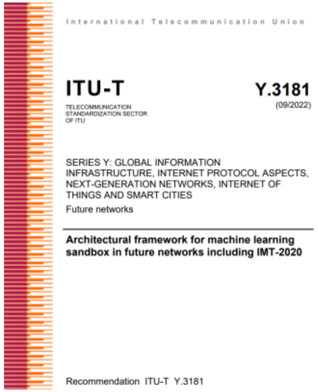


[GitHub - fwilhelmi/machine_learning_aware_architecture_wlans]

Wilhelmi, F., Barrachina-Muñoz, S., Bellalta, B., Cano, C., Jonsson, A., & Ram, V. (2020). [A flexible machine-learning-aware architecture for future WLANs]. IEEE Communications Magazine, 58(3), 25-31.

The ML Sandbox



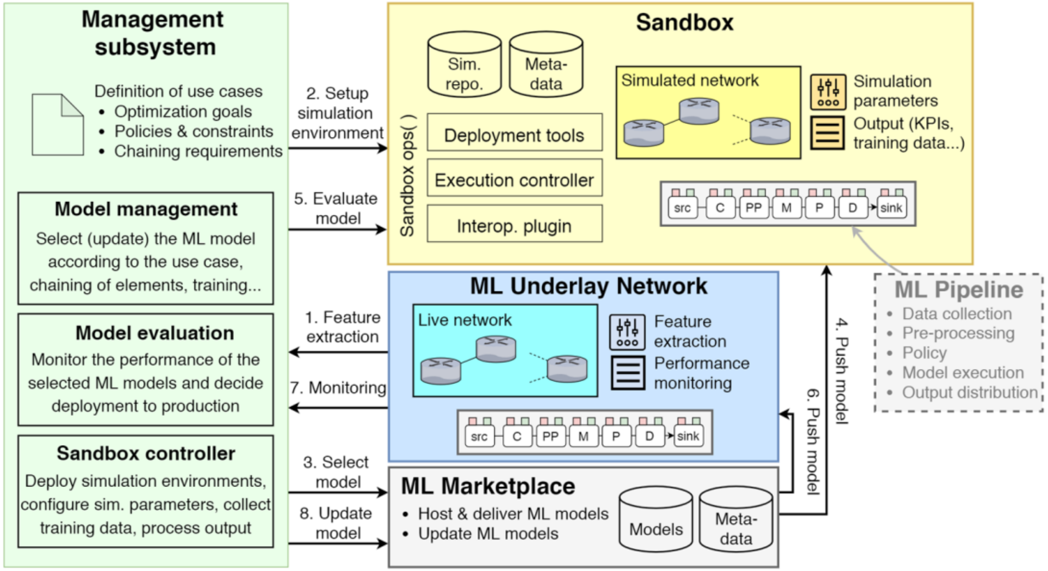


[Y.3181 : Architectural framework for machine learning sandbox in future networks including IMT-2020]

F. Wilhelmi, “Building reliability and trust with network simulators and standards”, ITU News Magazine No. 5, 2020

ML Sandbox & Network Digital Twins

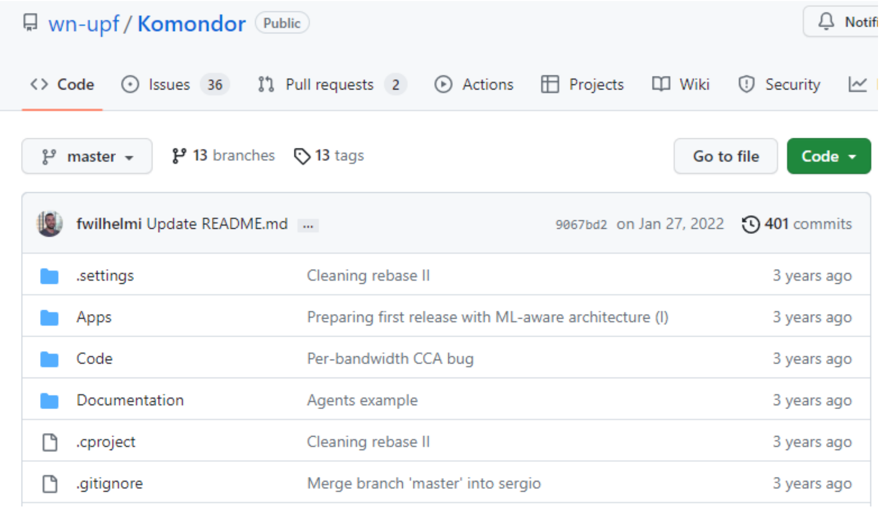
- Importance of network digital twinning in B5G communications
- Network simulators will play an important role
- Main responsibilities:
  - Train AI/ML mechanisms offline
  - Evaluate the performance of ML models
  - Generate data for training AI/ML models
- Challenges:
  - Interoperability
  - Flexibility, compatibility, and suitability
  - Intent-based operation
  - Closing the loop



Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ram, V., & Bellalta, B. (2021). Usage of network simulators in machine-learning-assisted 5G/6G networks. *IEEE Wireless Communications, 28(1), 160-166.*

Komondor

- Open-source: [GitHub - wn-upf/Komondor: Komondor Wireless Networks Simulator]
- A research and educational tool to simulate IEEE 802.11ax features and more
    - Low-complexity tool (simplified PHY/MAC)
        - Fast simulations – Very high deployments
        - Low-cost implementation of novel features, e.g., 11ax SR
        - Devise preliminary yet reliable results on specific 802.11 features
    - Includes “in-house” AI/ML operations
    - Suitable for creating datasets
- Good adoption in academia
    - Cited/used in >25 research works (source: Google Scholar)
    - Used in two editions of the ITU AI Challenge as a data generator [1], [2]



Simulator	Open-source	Source lang.	Complexity	GUI	11ax features	ML/based module
ns-3	Yes	C++	High	No[1]	Partial	No[2]
ns-2	Yes	C++/OTcl	Low	No[1]	No	No
OMNET++	No	C/C++	Medium	Yes	No	No
OPNET	No	C++	Medium	Yes	No	No
NetSim	No	Java	Low	Yes	No	No
Komondor	**Yes**	**C/C++**	**Low**	**No**	**Partial**	**Yes**

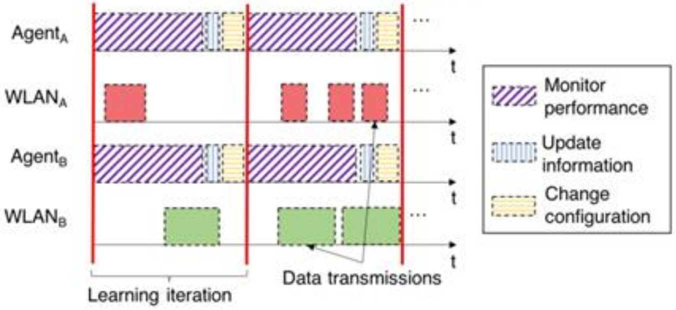
Wilhelmi, F. et al. (2021). Machine Learning for Performance Prediction of Channel Bonding in Next-Generation IEEE 802.11 WLANs. *ITU Journal on Future and Evolving Technologies, Volume 2 (2021), Issue 4 , Pages 67-79.*

Wilhelmi, F. et al. (2022). Federated Spatial Reuse Optimization in Next-Generation Decentralized IEEE 802.11 WLANs. *ITU Journal on Future and Evolving Technologies, Volume 3 (2022), Issue 2, Pages 117-133.*

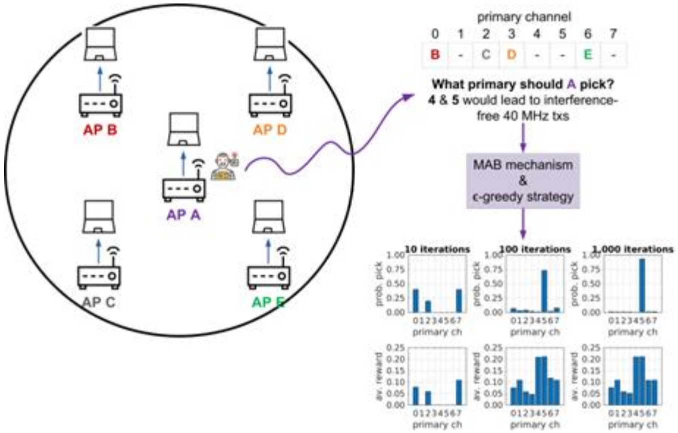
Barrachina-Munoz, S., Wilhelmi, F., Selinis, I., & Bellalta, B. (2019, April). Komondor: A wireless network simulator for next-generation high-density WLANs. *In 2019 Wireless Days (WD) (pp. 1-8). IEEE.*

Online agents operation in Komondor

(a) Agents embedded to APs



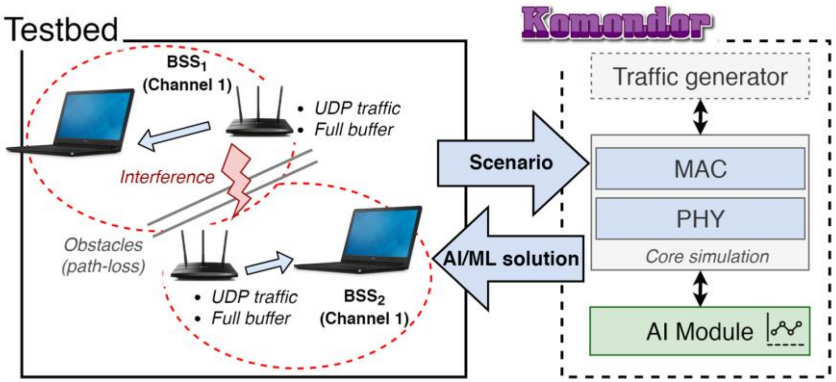
- The operation of AP and non-AP devices can be ruled by agents during simulations
- Agents run during the simulation
  - Monitoring performance (passive/reactive)
  - Internal operation (e.g., training/inference)
  - Submit configuration changes to devices
- Centralized, distributed, and <u>decentralized</u> modes
- An example:
  - Online learning is used by **AP$_A$** to decide the best primary channel
  - Scenario dependency (awareness of channels' statuses)
  - Transitory vs permanent regimes

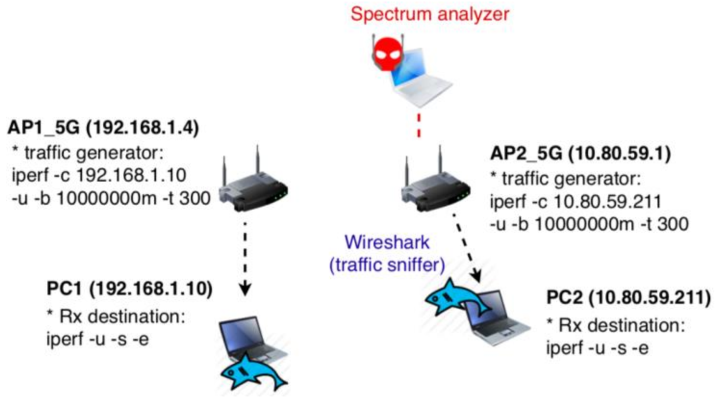


Komondor as ML Sandbox

Safe AIML model training*

- PoC on model training in Komondor
  - Characterize the real network in Komondor
  - Use AIML in Komondor to find a suitable network configuration in terms of tx power
  - Apply the configuration discovered in the simulator to the real network
- Testbed
  - 2 nearby BSSs (1 AP & 1 STA each) using the same channel
  - Full-buffer traffic generation at APs (*iperf*)
  - Traffic analysis at STAs (*Wireshark*)
  - Extra spectrum analyzer (*Aaronia*) to better understand the activity on the selected channel





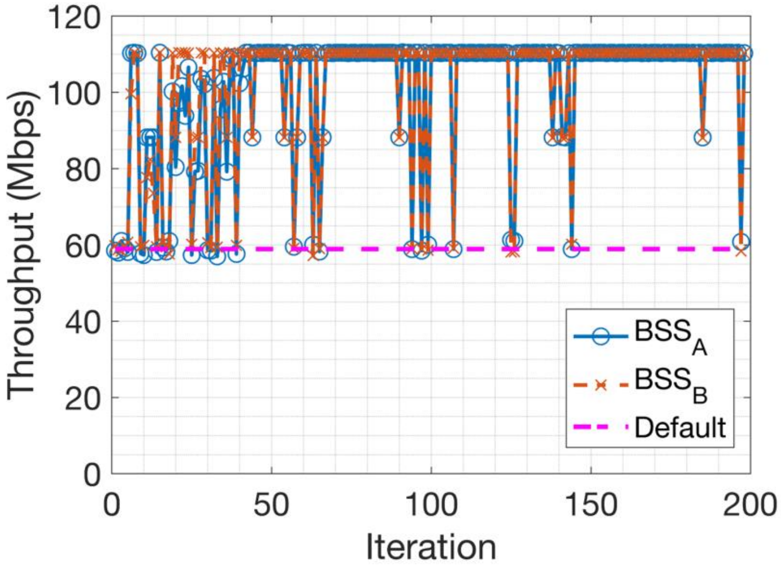
*Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ram, V., & Bellalta, B. (2021). [Usage of network simulators in machine-learning-assisted 5G/6G networks](). *IEEE Wireless Communications, 28(1), 160-166.*

Komondor as ML Sandbox

Safe AIML model training[♣]

Algorithm 1: Implementation of MABs in an OBSS

Function MAB ($\mathcal{P}$);
Input : $\mathcal{P}$: set of transmit power levels in $\{1, ..., K\}$
initialize: $t = 0$, $\forall k \in \mathcal{P}$, set $\hat{r}_k = 0$ and $n_k = 0$
while *active* **do**
  For each arm $k \in \mathcal{P}$, sample $\theta_k(t)$ from normal distribution $\mathcal{N}(\hat{r}_k, \frac{1}{n_k+1})$
  Play arm $k = \underset{1,...,K}{\text{argmax}}\ \theta_k(t)$
  Observe the throughput experienced Γ_t
  Compute the reward $r_{k,t}$
  $\hat{r}_{k,t} \leftarrow \frac{\hat{r}_{k,t} n_{k,t} + r_{k,t}}{n_{k,t}+2}$
  $n_{k,t} \leftarrow n_{k,t} + 1$
  $t \leftarrow t + 1$
end



- AIML method: online decentralized learning (MABs)
- Action space: {5, 7, 12, 17, 23} dBm

- On-simulation model training results
- 200 iterations = 200 seconds

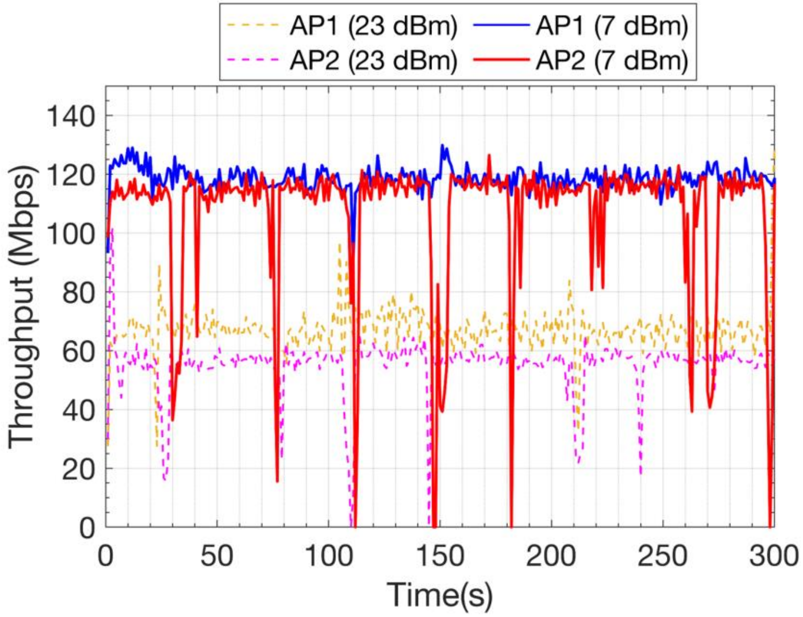
[♣]Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ram, V., & Bellalta, B. (2021). Usage of network simulators in machine-learning-assisted 5G/6G networks. *IEEE Wireless Communications, 28(1), 160-166.*

Komondor as ML Sandbox

Safe AIML model training[♣]

Tx Power	5 dBm	7 dBm	12 dBm	17 dBm	23 dBm
Probability	0.0804	0.7889	0.0452	0.0452	0.0503

- The best configuration observed in the simulator was 7 dBm (right-top Table)
- The right-bottom Figure shows the performance achieved when:
  a. Using the default configuration (maximum power = 23 dBm)
  b. Using the configuration discovered in the simulator (adapted power = 7 dBm)

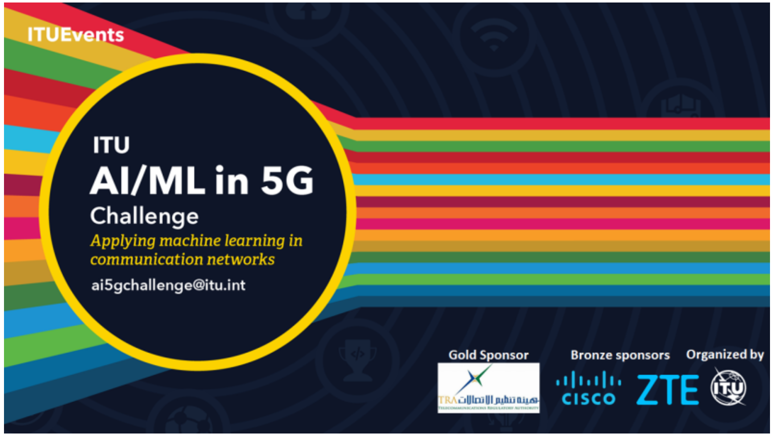


[♣]Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ram, V., & Bellalta, B. (2021). [Usage of network simulators in machine-learning-assisted 5G/6G networks](). *IEEE Wireless Communications, 28(1), 160-166.*

Komondor as ML Sandbox

Dataset generation for AIML model training

- **Problem:** Lack of data for training AIML models
  - Proprietary solutions / Competition
  - Privacy concerns
  - Cost of data gathering & processing
- **An appealing alternative:** Generate synthetic datasets with network simulators
- Komondor datasets:
  - 2020 ITU AI Challenge - IEEE 802.11ax Dynamic Channel Bonding (DCB) dataset♦
  - 2021 ITU AI Challenge - IEEE 802.11ax Spatial Reuse (SR) dataset♠



Wilhelmi, F., Góez, D., Soto, P., Vallés, R., Alfaifi, M., Algunayah, A., ... & Bellalta, B. (2021). Machine learning for performance prediction of channel bonding in next-generation IEEE 802.11 WLANs. *ITU Journal on Future and Evolving Technologies, Volume 2 (2021), Issue 4, Pages 67-79*.

Wilhelmi, F., Hribar, J., Yilmaz, S. F., Ozfatura, E., Ozfatura, K., Yildiz, O., ... & Bellalta, B. (2022). Federated spatial reuse optimization in next-generation decentralized IEEE 802.11 WLANs. *ITU Journal on Future and Evolving Technologies, Volume 3 (2022), Issue 2, Pages 117-133*

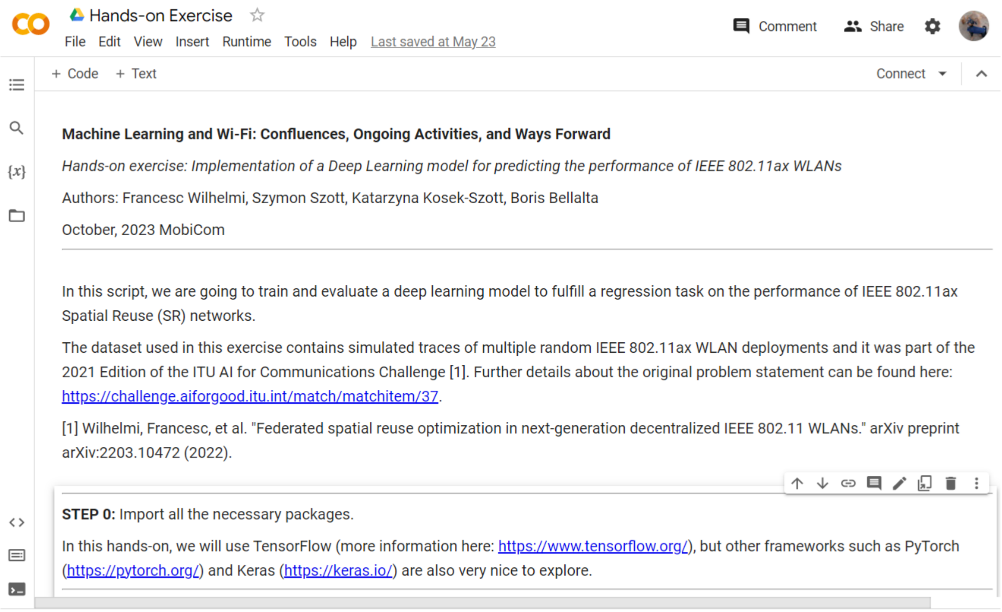
♦Wilhelmi, F. (2020). [ITU-T AI Challenge] Input/Output of project "Improving the capacity of IEEE 802.11 WLANs through Machine Learning" [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4106127

♠Wilhelmi, F. (2021). [ITU AI/ML Challenge 2021] Dataset IEEE 802.11ax Spatial Reuse (v1.3) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5656866

Hands-on exercise

Implement, Train & Evaluate an ML model

[Let’s go!]



Overview of tools & datasets

- ML frameworks, packages & libraries:
  - Pytorch: https://pytorch.org/
  - TensorFlow: https://www.tensorflow.org/
  - Keras: https://keras.io/
  - OpenAI Gym: https://github.com/openai/gym
- FL frameworks:
  - TensorFlow Federated: https://www.tensorflow.org/federated
  - PySyft: https://openmined.github.io/PySyft/
  - Flower: https://flower.dev/
- ML model energy consumption tracking:
  - CarbonTracker: https://github.com/lfwa/carbontracker
  - eco2AI: https://github.com/sb-ai-lab/Eco2AI
  - CodeCarbon: https://github.com/mlco2/codecarbon

Outline

Part 1: **Introduction. Why Wi-Fi may want to adopt AI/ML?** (30 mins) - Katarzyna
- Wi-Fi overview: A 30-year path from IEEE 802.11b to IEEE 802.11be, and beyond.
- Open challenges in Wi-Fi: Dealing with complexity and uncertainty.
- Requirements of next-gen WiFi networks.

Part 2: **A primer on AI/ML** (45 min) - Szymon & Boris
- Concepts, definitions, and overview of the main ML types (supervised, unsupervised, and reinforcement learning), including Wi-Fi examples.
- Popular ML techniques and paradigms: deep learning, reinforcement learning, online learning, federated learning.
- Deployment options: architecture, data handling, marketplaces.

Break (10:00-10:30)

Part 3: **Multi-Armed Bandits for Responsive Wi-Fi networks** (45 mins) - Boris & Szymon
- Multi-Armed Bandits: exploitation-exploration trade-off; e-greedy, Thompson Sampling, UCB.
- Examples: Channel Selection; AP selection;
- Reinforced-lib + Example (MCS selection).

Part 4: **Wi-Fi & ML: Practical considerations** (45 mins) - Francesc
- Wi-Fi & ML: A two-sided relationship
- Adoption of ML in Wi-Fi
- Hands-on Exercise II: Predicting the performance of Wi-Fi through Deep Learning

Part 5: **Open challenges, future research directions, and summary** (15 mins) - Katarzyna
- Open challenges and future research directions; Synergies with other disruptive technologies.
- Current AI/ML trends: LLM and Generative AI.
- Summary and takeaways.

5
Open challenges, future research directions, and summary

Research community generates new ideas…

- **Optimizing new 802.11 features**
  - Not yet well-known how ML can improve the performance of existing features
  - What about new ones? Multi-link operation, multi-AP coordination
- **Joint parameter optimization**
  - Complex interplay between different features
  - E.g., transmit power and carrier sensing threshold, channel/link selection
- **Distributed and transfer learning; hierarchical learning**
  - Training and sharing/deploying models in a distributed way
  - Reuse of knowledge between features

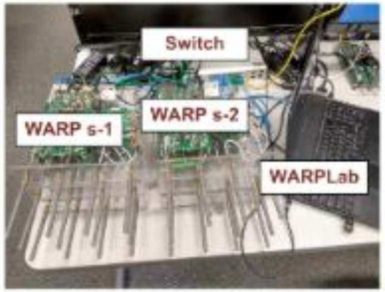
Datasets

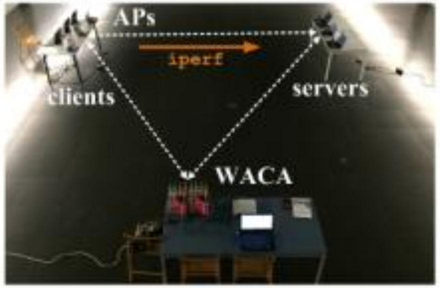
- The existence of open-source and standardized datasets is essential for training and comparing ML-based algorithms.
- Very few datasets are currently available.
- Examples of datasets:
    - Herzen et al. provide a dataset to **predict throughput** based on basic performance metrics (e.g., **received power, channel width**) collected in a small testbed.
    - Karmakar et al. provide the IEEE 802.11ac performance dataset that contains information regarding **normalized throughput** achieved under five **link configuration parameters** (i.e., channel bandwidth, MCS, guard interval, MIMO, and frame aggregation) and the channel quality measured as SNR.
    - Chen et al. provide a dataset for **long-time user load prediction** in large-scale WiFi system.
- It would be beneficial to have standardized datasets, standardized means for sharing datasets (preserving privacy), and standardized procedures for data collection (to build new datasets.)

More info
in our survey:

New Simulation & Experimental Tools

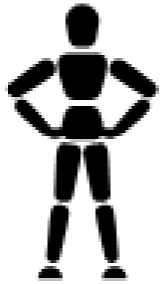
- Wi-Fi networks are becoming extremely complex
  - Many new features and different scenarios
- We need (new) evaluation tools (and metrics) to obtain reliable and reproducible performance results: analysis, simulations, testbeds
  - Include AI/ML support
- The case of ns-3:
  - Full TCP/IP stack, Wi-Fi, community validated, open-source
  - Not all 11ax features yet implemented; some 11be ones in development (developing new features requires expertise and time)
  - **AI/ML support:** ns3-gym: OpenAI Gym integration, **Reinforced-lib**



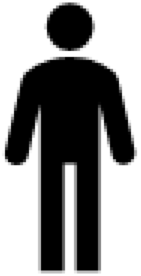


AI/ML in Practice

- **Most of current results in the literature come from simulations:**
  - Many simplifying assumptions:
    - Controlled (simple) scenarios, discrete time (iterations)
    - Instantaneous and noise-free data: ideal ‘devices’ (no need to calibrate data), no estimation errors, etc.
    - No ML overheads (data and model exchange) usually considered
  - **The ‘gap’ with performance results from real implementations could be significant**



- **Engineering effort to set-up and test AI/ML solutions in practice is still required**
  - Suitable KPIs are required, including how and when to measure them
  - When should AI/ML agents be triggered?
    - Instability vs slow reaction;
    - Should ‘thresholds’ be used to activate ML mechanisms?



AI/ML in Practice

- Performance gains in practice
  - While AI/ML MAC/PHY enhancements affecting specific mechanisms may be easy to evaluate (link parameter selection, CW adjustment, beamforming, etc.), **it may not be easy to assess other AI/ML performance gains** since they may appear only in particular situations (e.g., high loaded scenarios).
    - Example: network-controlled client device association and band steering (the AP uses AI/ML to determine when it is better to move one client from one band to another)
- Fairness issues between AI/ML-enabled devices and legacy devices
- Privacy is one of the most important future challenges

New research directions

Generative-Adversarial-Network (GAN)

- GAN-based wireless channel modeling
  - GAN is used to "**autonomous wireless channel modeling without complex theoretical analysis or data processing**"
  - "GAN is trained by raw measurement data to reach the Nash equilibrium of a MinMax game between a channel data generator and a channel data discriminator"
  - "The resulting channel data generator is extracted as the target channel model"
- GAN-based model-free resource allocation
  - GAN "is used to pre-train the deep-RL framework using a mix of real and synthetic data, thus creating an **experienced deep-RL framework** that has been exposed to a broad range of network conditions."
  - "deep-RL framework applied to a multi-user orthogonal frequency division multiple access (OFDMA) resource allocation system."
- Call for papers regarding generative AI interplay with 5G/6G wireless networks

New research directions

Large Language Model (LLM)

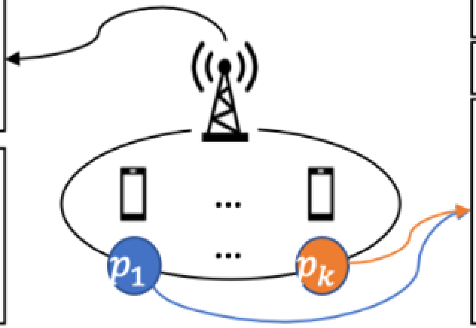
- LLM-based hardware development
  - “LLMs are used in **FPGA-based hardware development** for advanced signal-processing algorithms in wireless communication and networking”
- LLM-based anomaly resolution and optimization of mobile networks
  - Allows “capturing different KPIs of the network and the interactions between various network configuration parameters”
  - Anomaly detection based on “**troubleshooting tickets accumulated over time**”
- Collaborative multi-agent LLMs (intent driven networking)
  - Existing problem: huge parameter sizes
  - **Multi-agent light on-device LLMs collaboratively planning and solving tasks to achieve network goals**
    - Semantic communication with task specific knowledge
    - Reduced model size and computing costs
  - GPT-4 was used to generate exemplary results (next slide)

New research directions

Network power consumption with LLM

Scenario: Consider the downlink channel with one base station and k users. The noise level is the same for all users and equals to n dB. The bandwidth is b kHz. The channel gain vector is g. The initial transmission power vector is p.

Game rule: in each round, each user should choose its power so that its transmission rate is no less than its minimum rate and the difference (always non-negative) of its current transmitting power w.r.t. its intial power is minimized. There are x rounds for each user to choose the transmitting power.

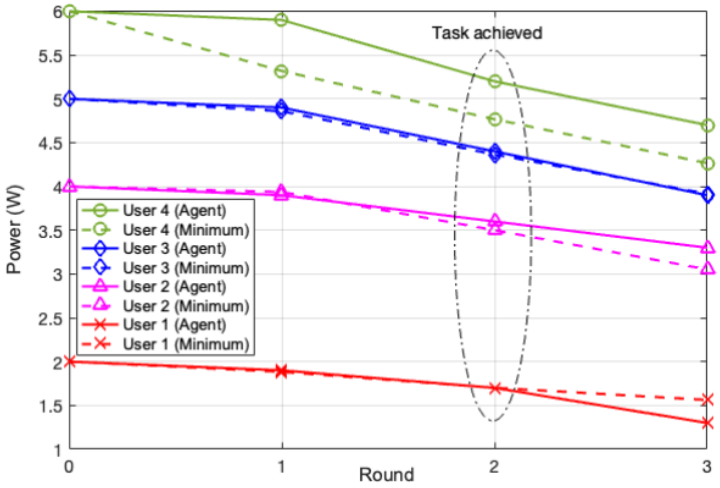


Network goal: reduce the total power by Δp W at least.

User condition: minimum transmission rate at r kbps.

It is currently round x_t. You are user k_i.
In last round, transmitting power including yourself is: p'
When you calculate your transmission rate, you can use $r_k = B\log(1+\frac{p_k g_k}{n^2+\Sigma_{i\neq k} p_i g_i})$
How will you choose your transmitting power in next round?
Your reply should be like 'my transmitting power is ... W'

"This use-case demonstrates that an LLM (GPT-4) can perform mathematical reasoning in a wireless communication-based problem, in order to achieve a global power saving and individual transmission rates targets"



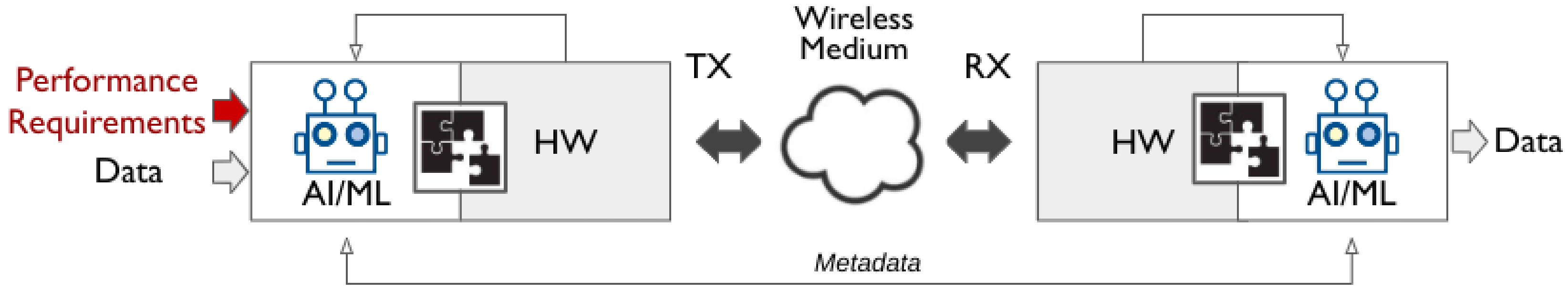
Source: Zou, Hang, et al. "Wireless Multi-Agent Generative AI: From Connected Intelligence to Collective Intelligence." *arXiv preprint arXiv:2307.02757* (2023).

Summary & Take-home Messages

1. ML is a set of **general techniques** to create models for **making predictions**. They can be used to empower autonomous ‘agents’ to find suitable configurations in particular scenarios.
2. Given the decentralized nature of 802.11 deployments, the use of **RL techniques** looks like the natural choice to **dynamically improve** several features such as channel selection, spatial reuse, channel access parameters, etc.
3. Performance gains in practice may not be so obvious
    a. AI/ML enhanced 802.11 networks should improve user’s **Quality of Experience** by providing a fast response to challenging situations.
4. The **802.11 architecture should be updated** to effectively support AI/ML by adding the necessary interfaces, procedures, and data formats

MLDR - Machine Learning Defined Radio

- **Chist-ERA project**: 2024-2027
- **Partners**: AGH, Supélec, U. Oulu, UPF, with the support from *Nokia, Interdigital, NetAI, DeepSig, Canon Research.*
- **Goal**: *"In this project, we aim to build a new, clean-slate AI/ML-Driven Radio (MLDR) interface. This new MLDR interface will learn to communicate by selecting and configuring the set of communication protocols and functionalities that better suit every particular use-case and scenario, thus satisfying the aforementioned hard performance requirements and efficiently using the available spectrum resources."*



Thank you!

https://mlwifitutorial.github.io/