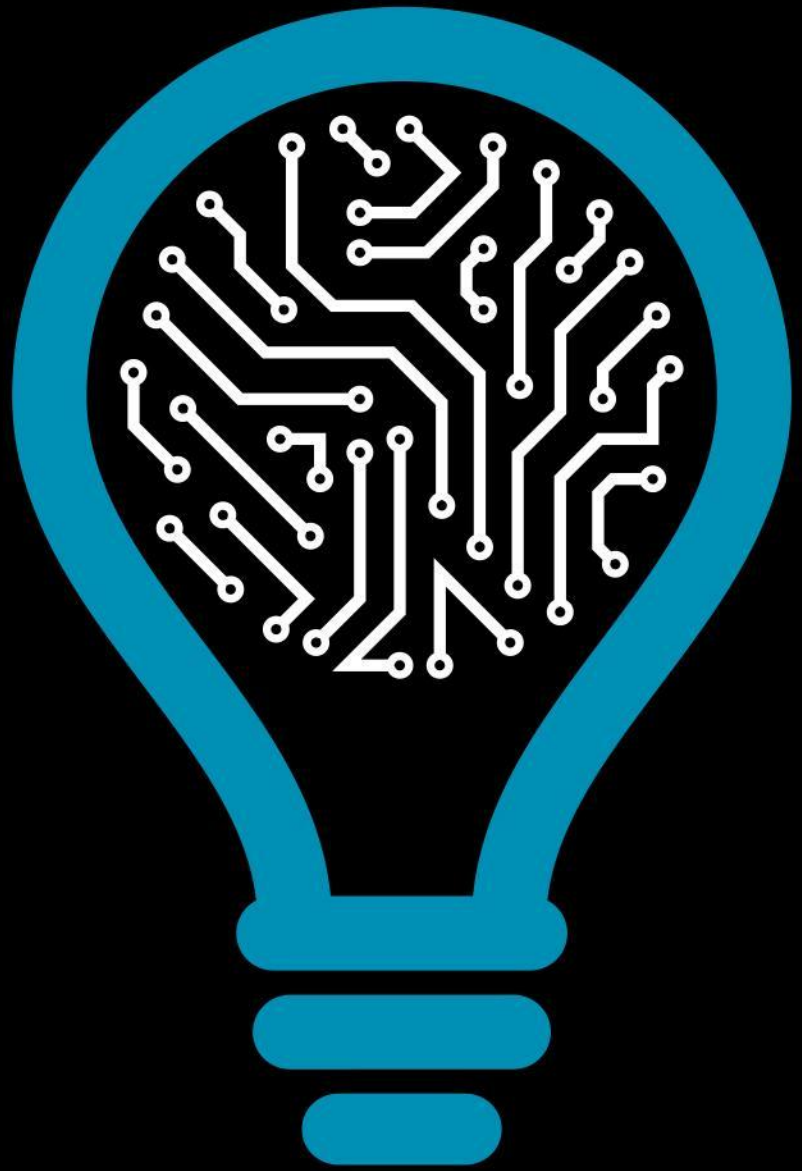
Using Machine Learning to Improve the Performance of Flying Networks



- Francisco Tuna de Andrade
- Pedro Miguel Ferraz Nogueira da Silva

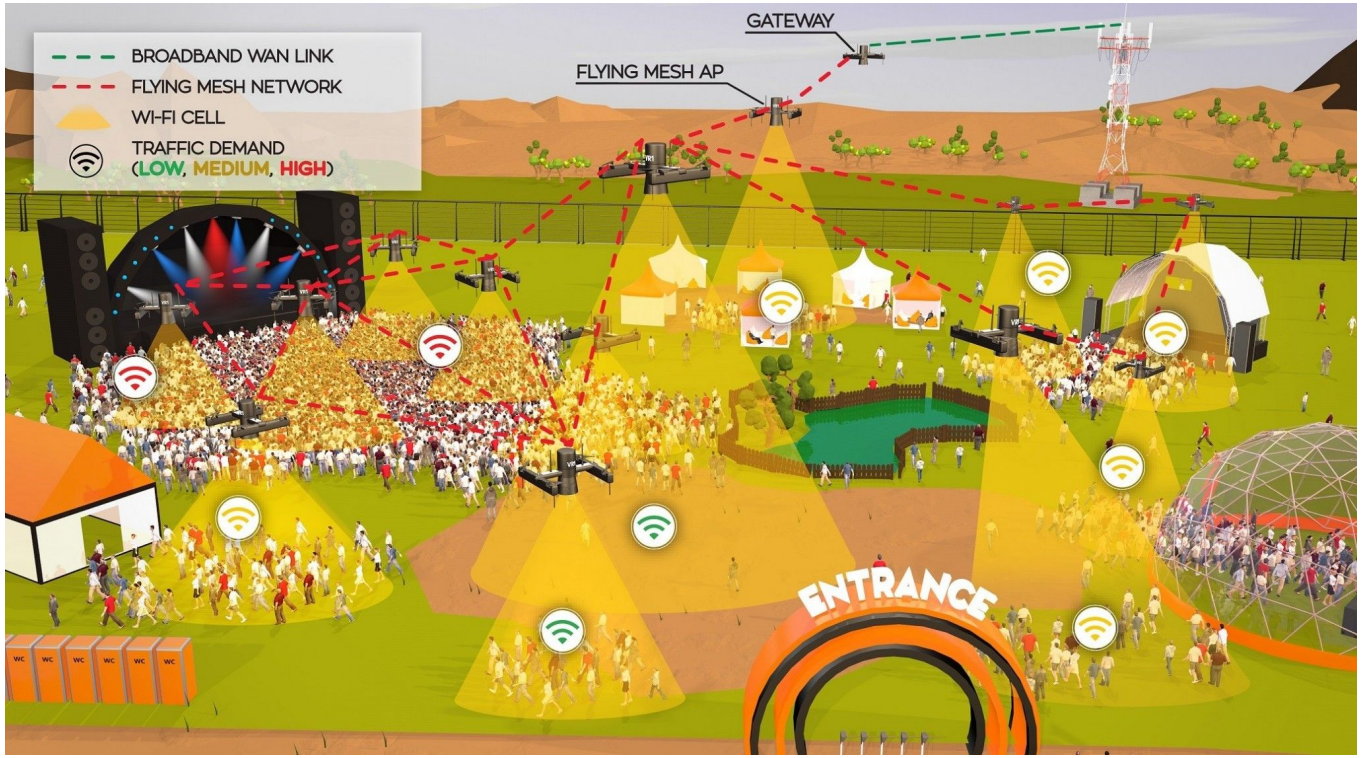
3/08/2018

INESCTEC

INSTITUTO DE ENGENHARIA DE SISTEMAS E COMPUTADORES, TECNOLOGIA E CIÊNCIA

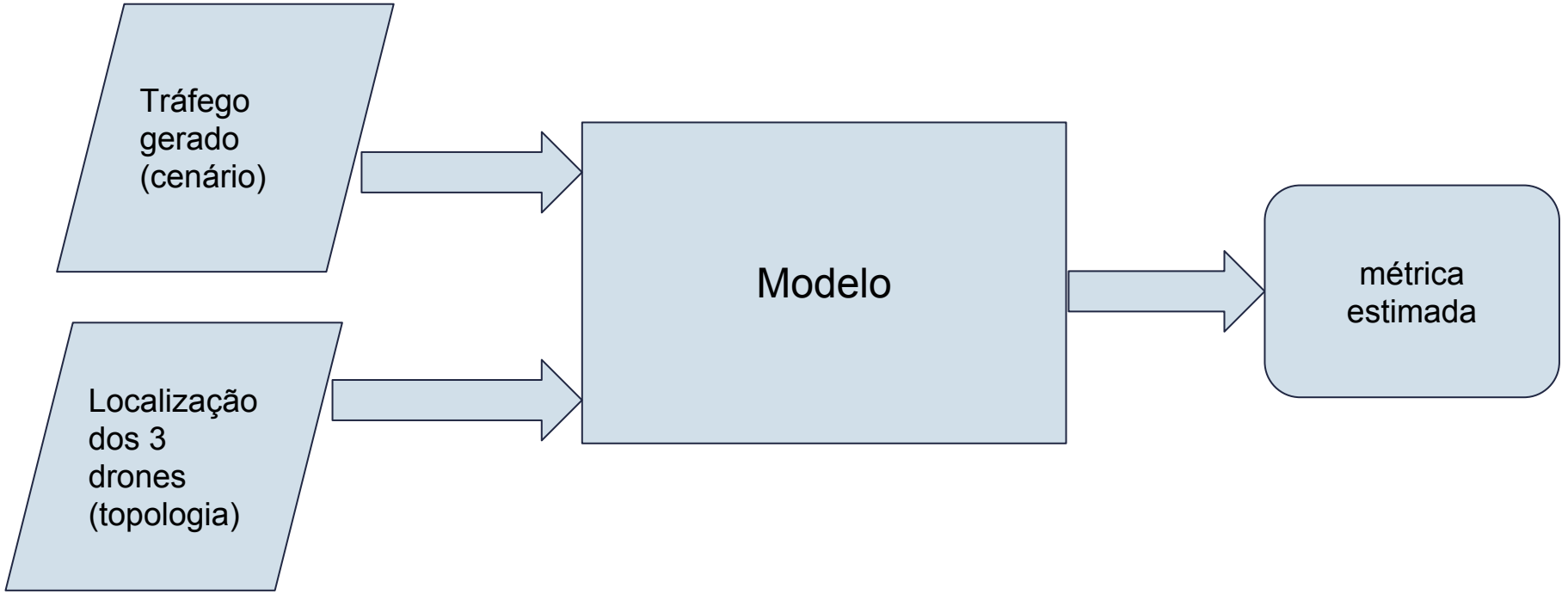
Enquadramento

- Uso de redes baseadas em drones (UAV's) como resposta a eventos temporariamente lotados (TCE's)

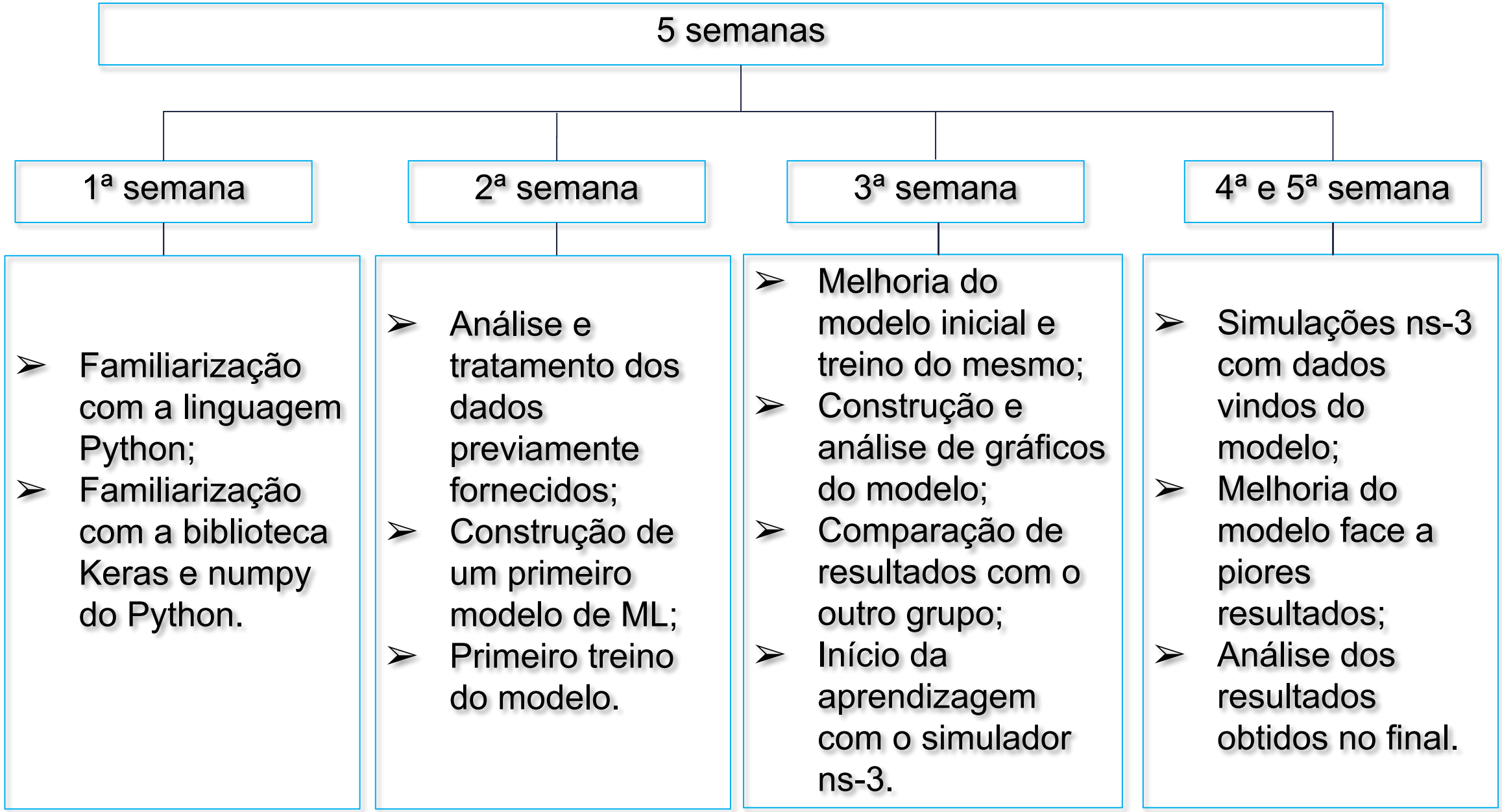


Objetivos

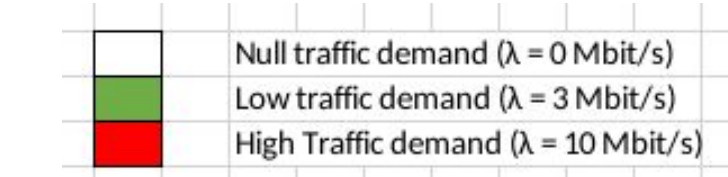
- Construção de rede neuronal que preveja as métricas (throughput, delay ou pdr) de uma rede UAV

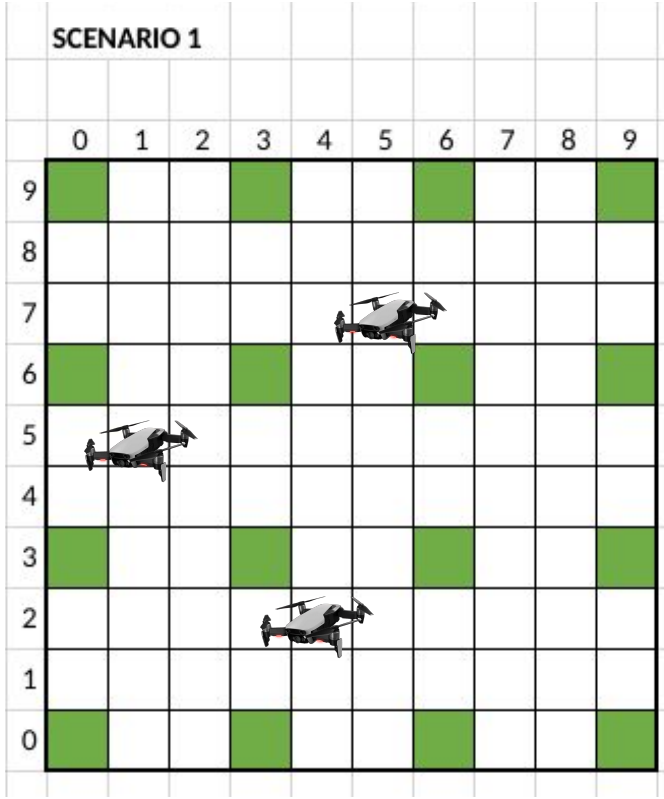


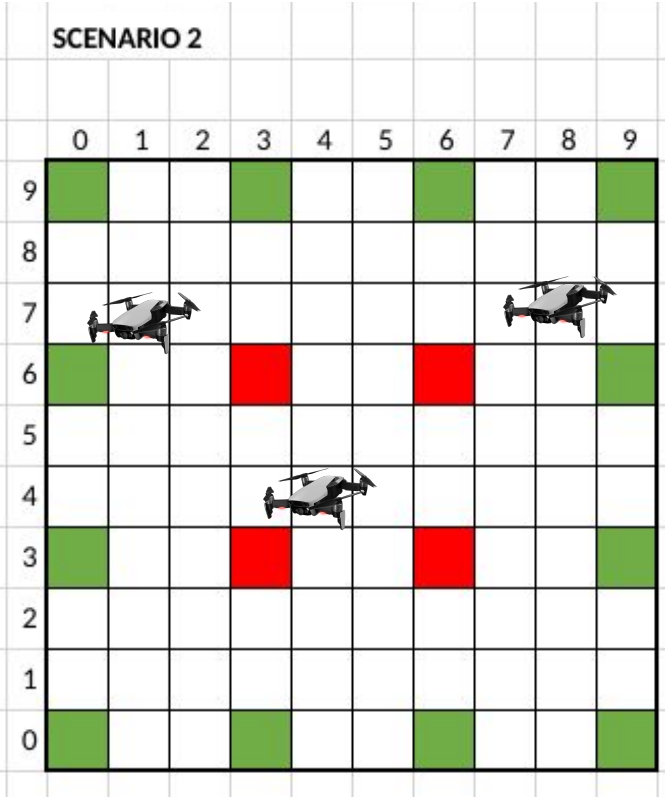
Trabalho desenvolvido

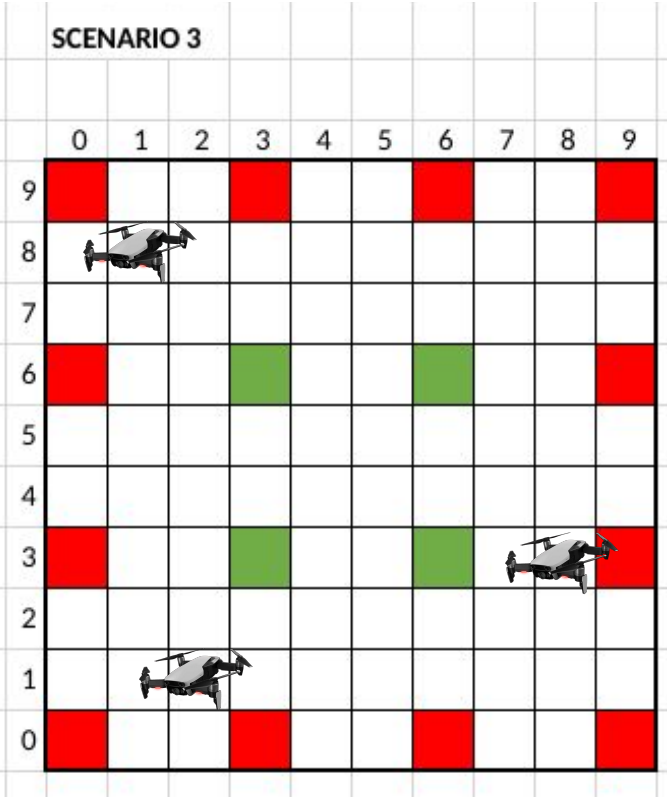


Dataset









- Posição dos 3 Drones;
- Altitude de 10 metros;
- *WifiCellRange* de 100 metros
- *WifiChannelNumber* : {36,40,44}
- Métricas a analisar (*meanRxBitrateMbps,meanDelayMs,meanJitterMs,meanPdr*)

Processamento dos dados

Linha de comandos - python -i build_model.py

```
>>> input_train
array([[[[ 3.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  3.        ],
         [ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ],
         [ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ],
         ...,
         [ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ],
         [ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ],
         [ 3.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  3.        ]],

        [[ 3.02953607,  3.06634492,  2.80236837, ...,  0.43139197,
           0.16931388,  0.07336994],
         [ 2.43830285,  2.4803205 ,  2.60391312, ..., -0.02806965,
          -0.37829206, -0.51786329],
         [ 1.84706962,  1.89600158,  2.03895751, ..., -0.37829206,
          -0.8641997 , -1.10909652],
         ...,
         [-1.10909652, -0.8641997 , -0.51786329, ...,  0.94374563,
           0.7373873 ,  0.66460316],
         [-1.70032975, -1.10909652, -1.10909652, ...,  1.31438002,
           1.31438002,  1.25583639],
         [-1.10909652, -1.10909652, -1.70032975, ...,  1.25583639,
           1.84706962,  1.84706962]]],


       ...,

       [[[ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ],
         [ 0.        ,  3.        ,  0.        , ..., 10.        ,
          10.        ,  0.        ],
         [ 0.        ,  0.        ,  0.        , ...,  3.        ,
          10.        ,  0.        ],
         ...,
         [ 0.        ,  0.        ,  3.        , ...,  0.        ,
           0.        ,  0.        ],
         [ 0.        ,  3.        ,  0.        , ...,  3.        ,
           0.        ,  0.        ],
         [ 0.        ,  0.        ,  0.        , ...,  0.        ,
           0.        ,  0.        ]],

        [[ 7.14390291,  6.14881563,  5.19491291, ...,  4.4517146 ,
           5.19491291,  6.14881563],
         [ 5.47674311,  4.90466431,  3.80958332, ...,  2.9295496 ,
           3.80958332,  4.90466431],
         [ 3.80958332,  3.80958332,  2.54694432, ...,  1.48481447,
           2.54694432,  3.80958332],
         ...,
         [-4.52621566, -2.85905587, -1.19189607, ...,  0.18922433,
           1.48481447,  2.9295496 ],
         [-4.52621566, -2.85905587, -1.19189607, ...,  1.48481447,
           2.54694432,  3.80958332],
         [-2.85905587, -2.16849567, -0.79833303, ...,  2.9295496 ,
           3.80958332,  4.90466431]]]])
>>> throughput_train
array([2.689776  , 2.952488  , 2.928736  , ..., 2.805651  , 2.15420875,
       2.665089  ])
>>>
```

Fig 2: Dados de treino

```
#Global Variables
SCENARIO_ROWS = 10
SCENARIO_COLUMNS = 10
SCENARIO_TOPOLOGIES_NO = 200
SCENARIOS_NO = 10
TOPOLOGIES_TRAINING = 128
TOPOLOGIES_VALIDATION = 32
TOPOLOGIES_TESTING = 40
#SCENARIOS_TRAINING = [1, 2, 4, 6, 7, 10]
#SCENARIOS_VALIDATION = [5, 9]
SCENARIOS_TRAINING = [1, 2, 4, 6, 7, 10, 5, 9]
SCENARIOS_VALIDATION = []
SCENARIOS_TEST = [3, 8]
DIVISION_BY_TOPOLOGIES = 0
DISTANCE_ENCODING = 1
NORMALIZE_DATA = 1
USE_TRANSFORMATIONS = 1
CHANNELS_LAST = 0
VALIDATION_SPLIT = 0
USE_CALLBACKS = 0
TEST_RESULTS = 1
```

Fig 3: Tipos de settings usados

Camadas do Modelo

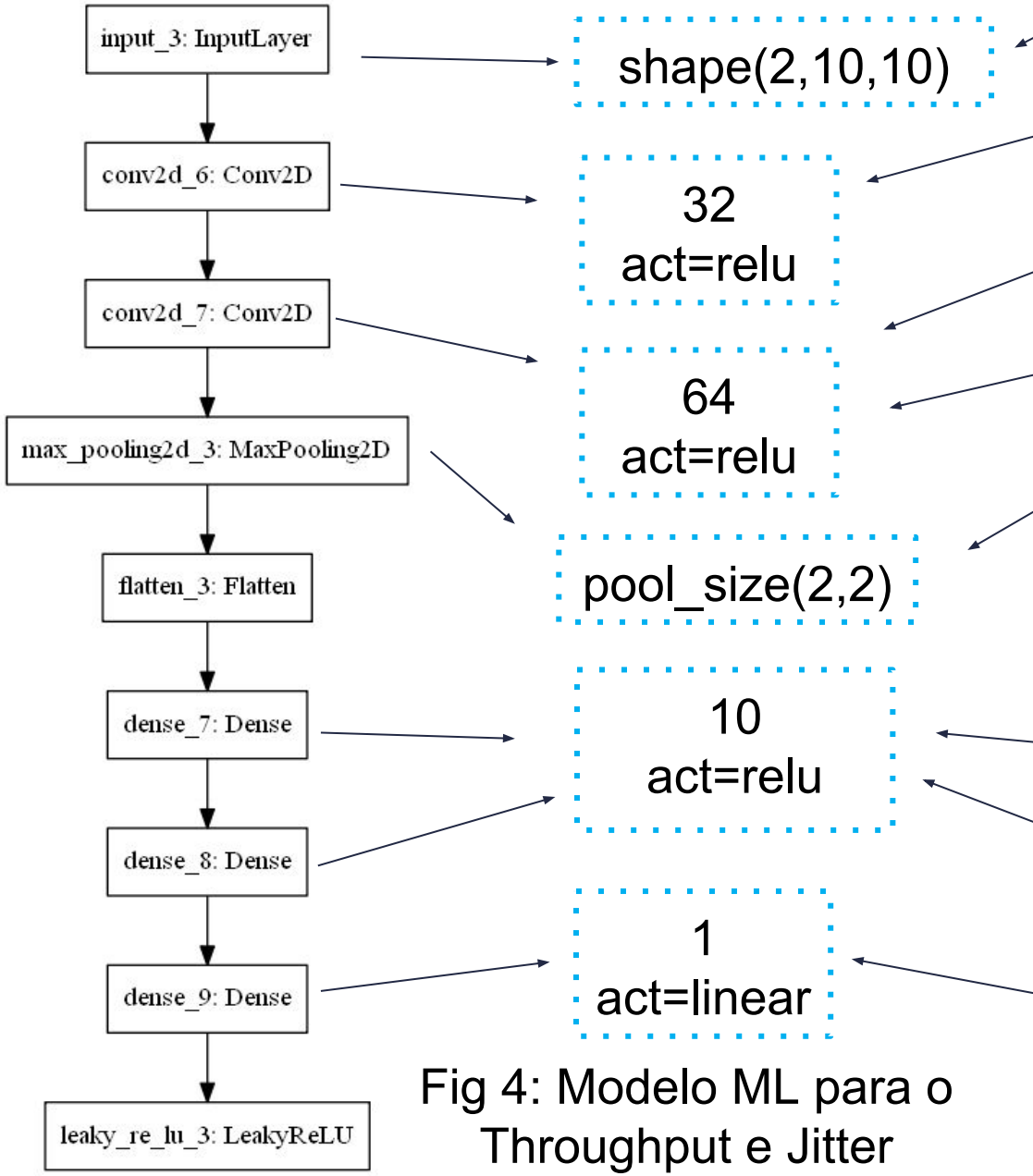


Fig 4: Modelo ML para o Throughput e Jitter

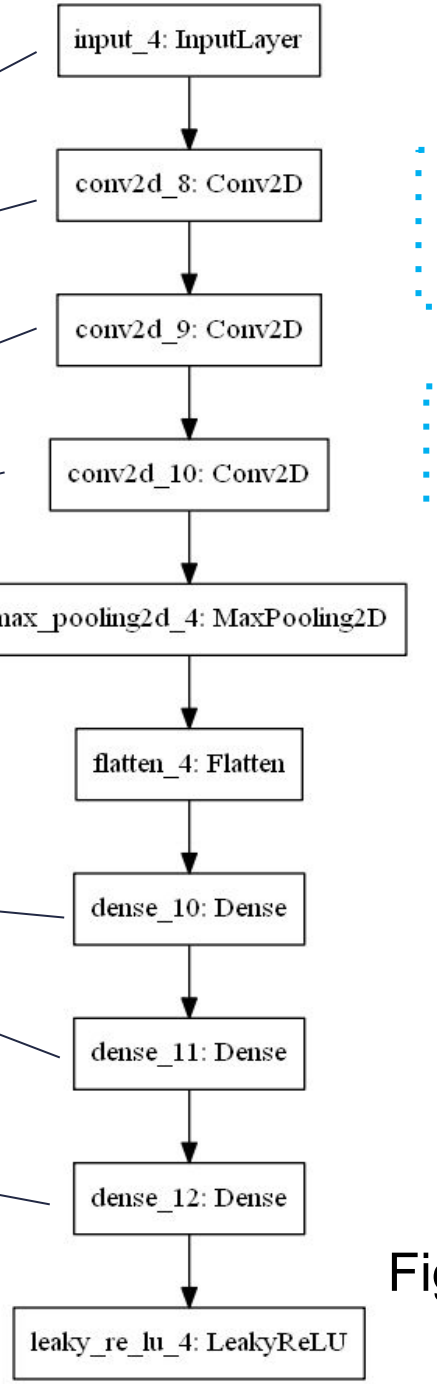
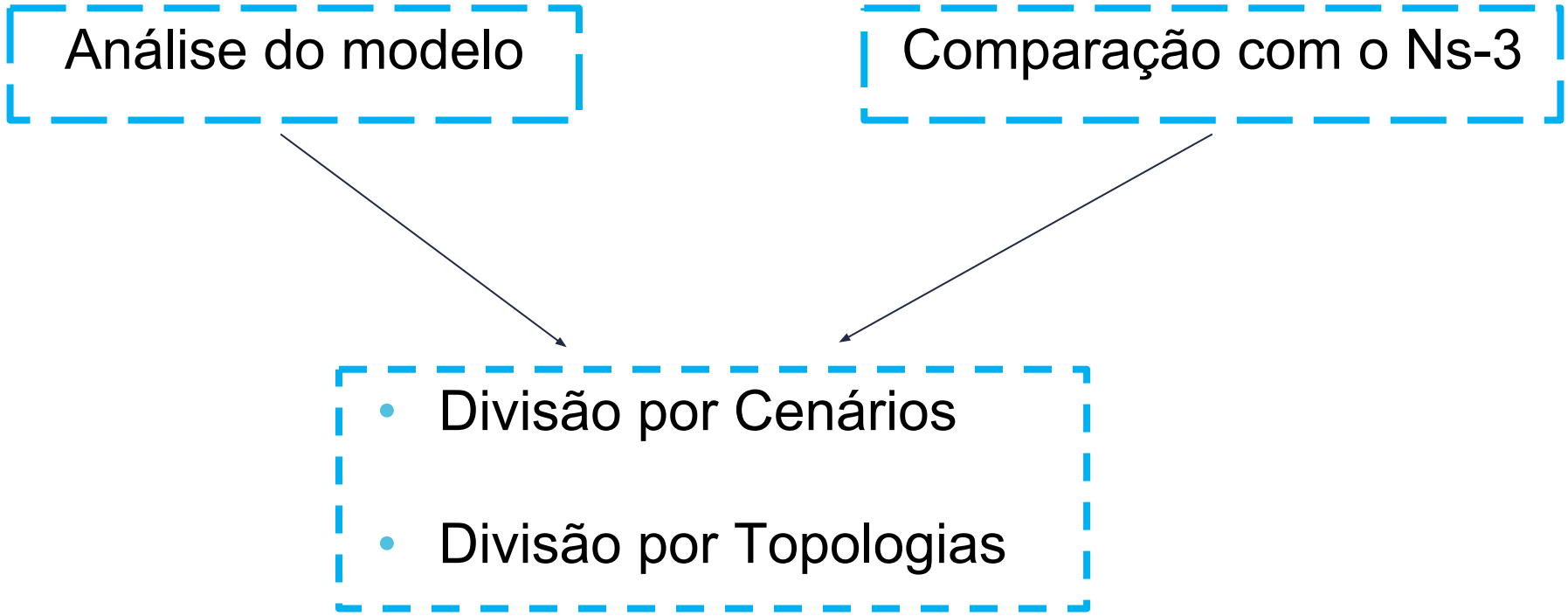


Fig 5: Modelo ML para o Delay e Pdr

Resultados obtidos



Resultados obtidos

- Análise do modelo

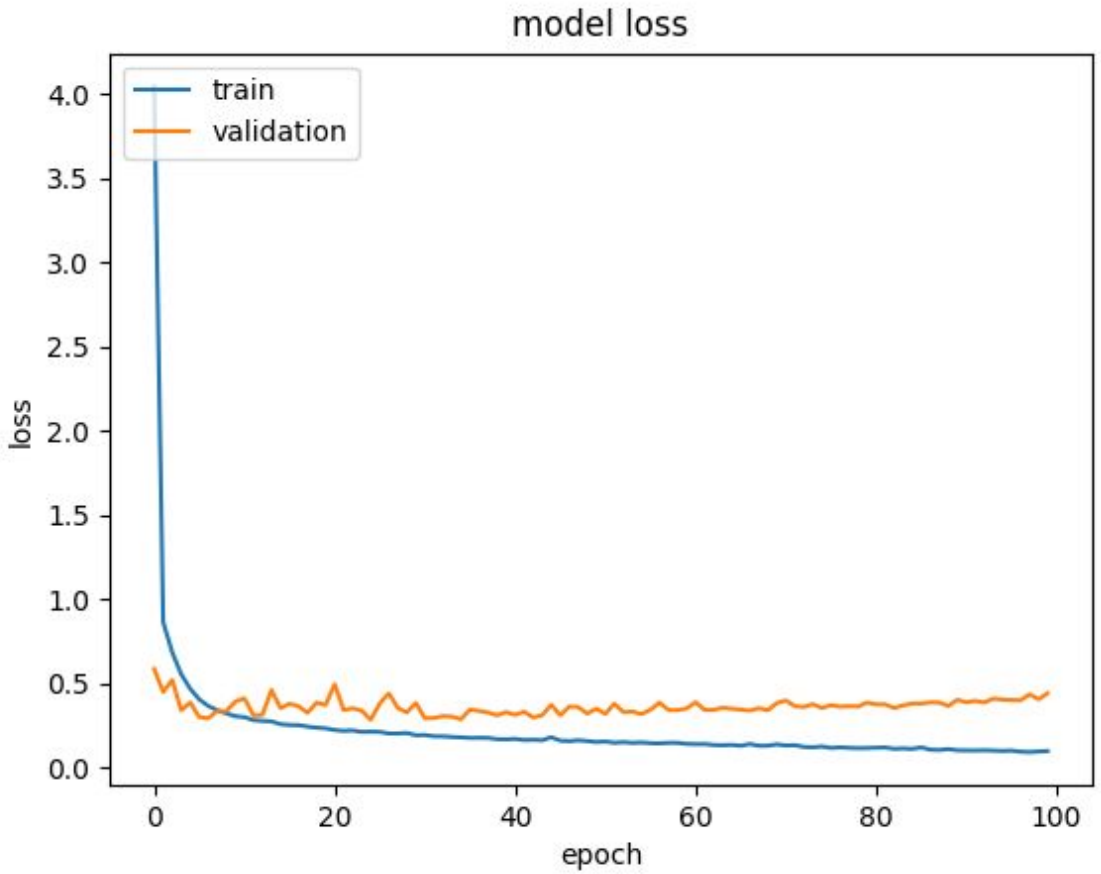
Train/Test		12800/3200					12800/3200		
Division		**Scenarios**					**Topologies**		
		MSE	**MAE**	**MAPE**			**MSE**	**MAE**	**MAPE**
	Throughput	0.833	0.762	17.619		**Throughput**	0.176	0.296	7.976
	Delay	20609.9	388.687	41.118		**Delay**	39241.7	147.179	18.929
	PDR	0.041	0.172	29.119		**PDR**	0.003	0.041	6.46

Fig 6: Análise de treino do modelo

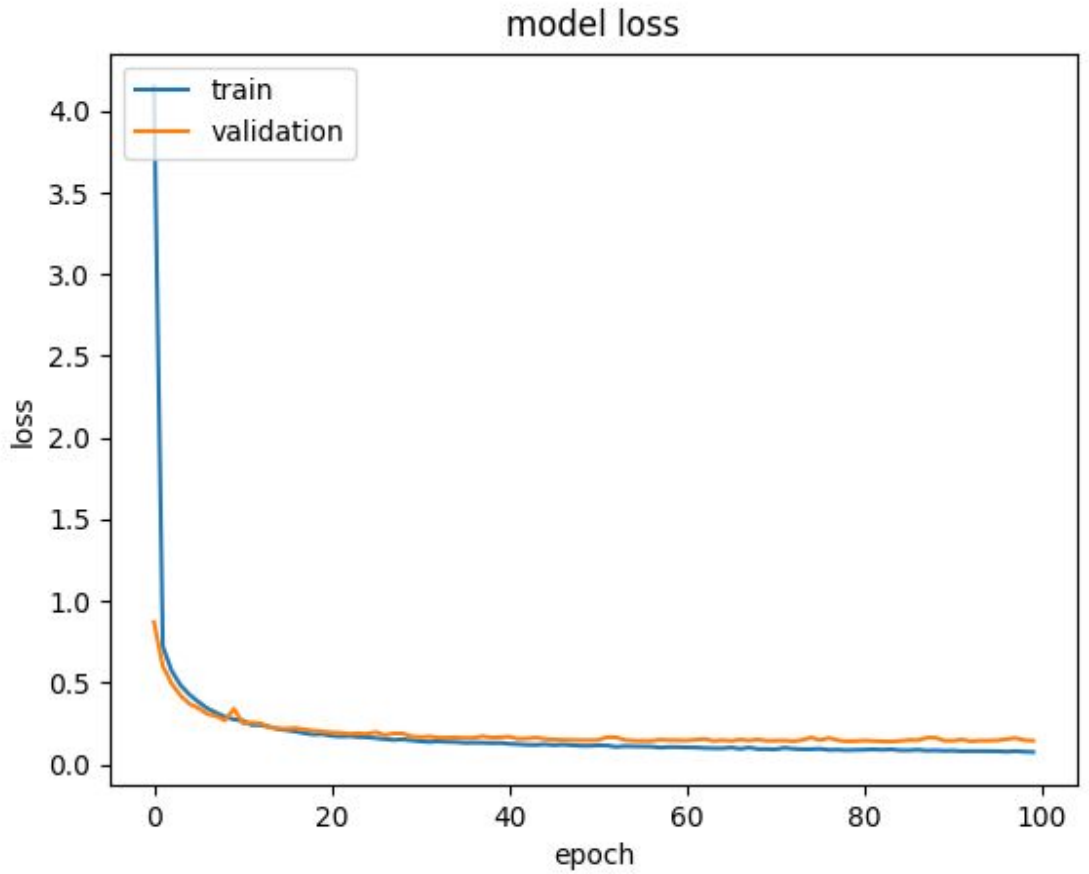
Resultados obtidos

- Análise do modelo

Divisão por Cenários:



Divisão por Topologias:

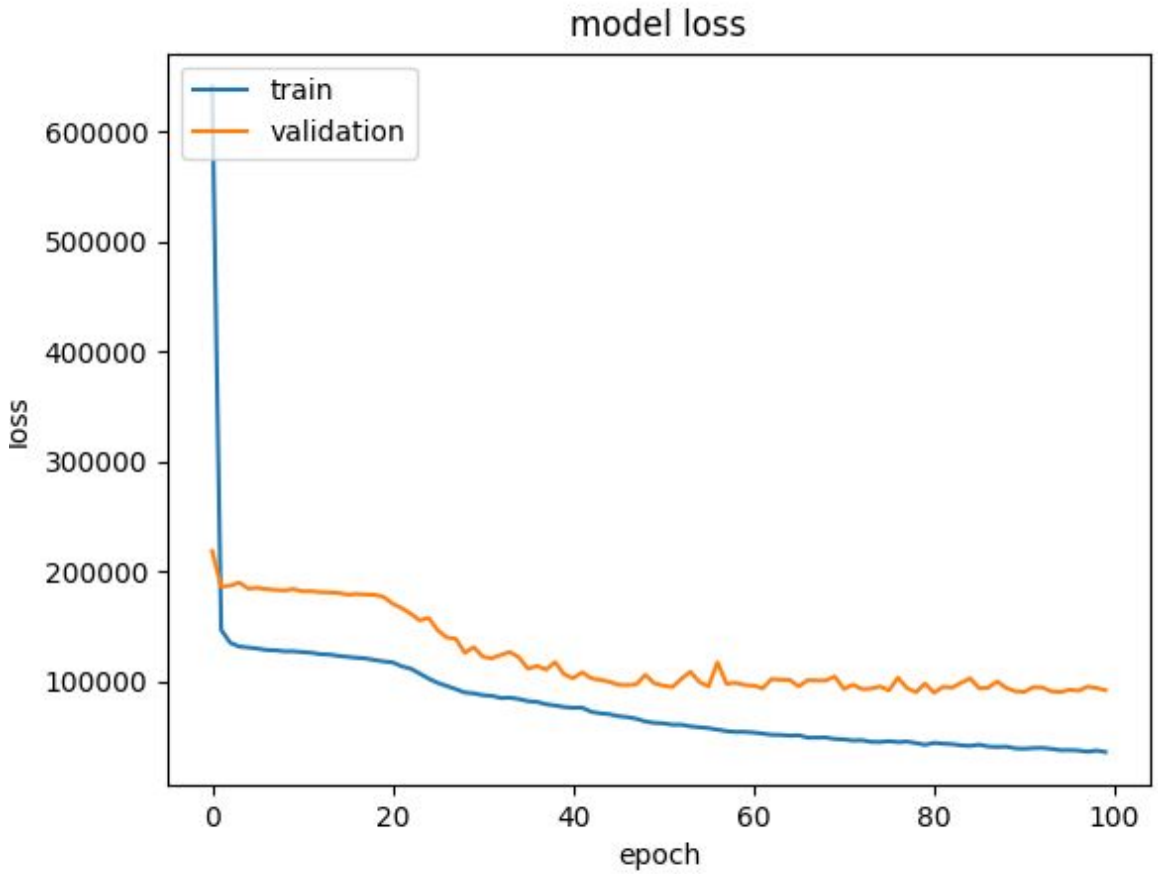


Throughput Loss

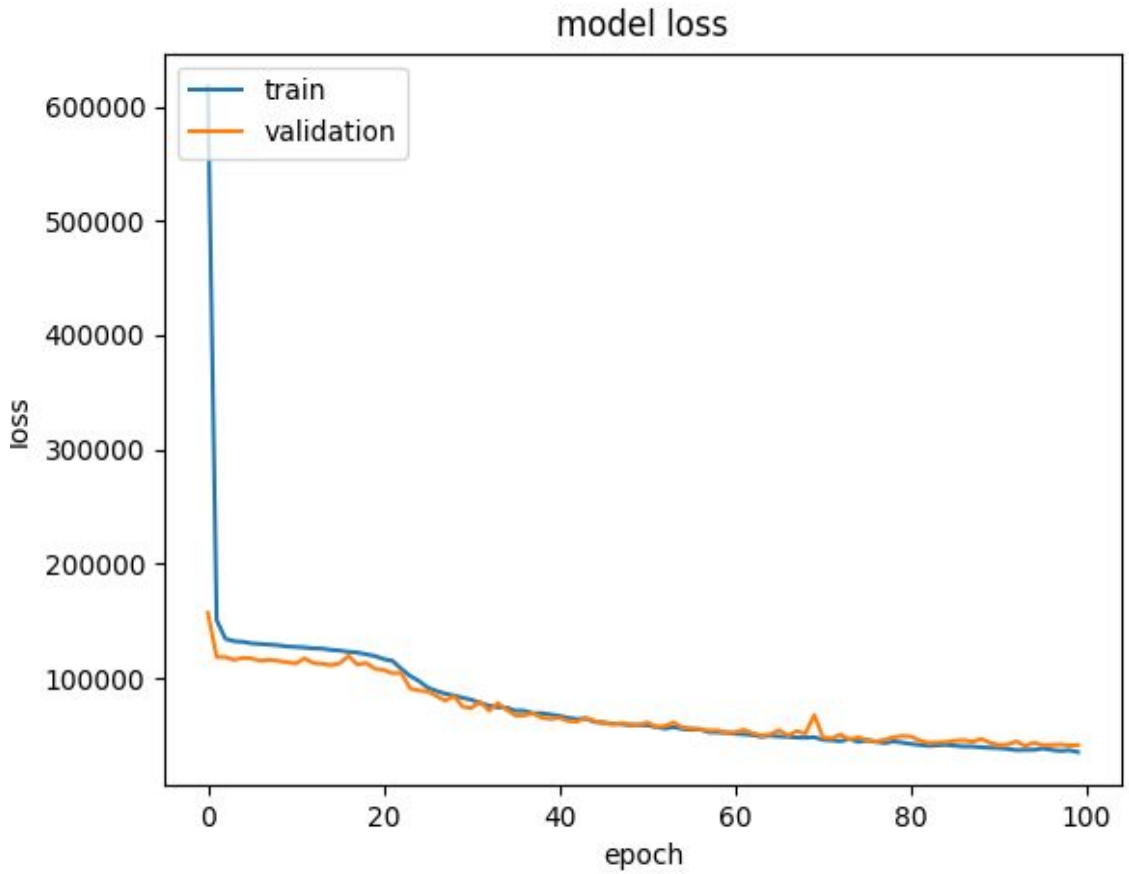
Resultados obtidos

- Análise do modelo

Divisão por Cenários:



Divisão por Topologias:

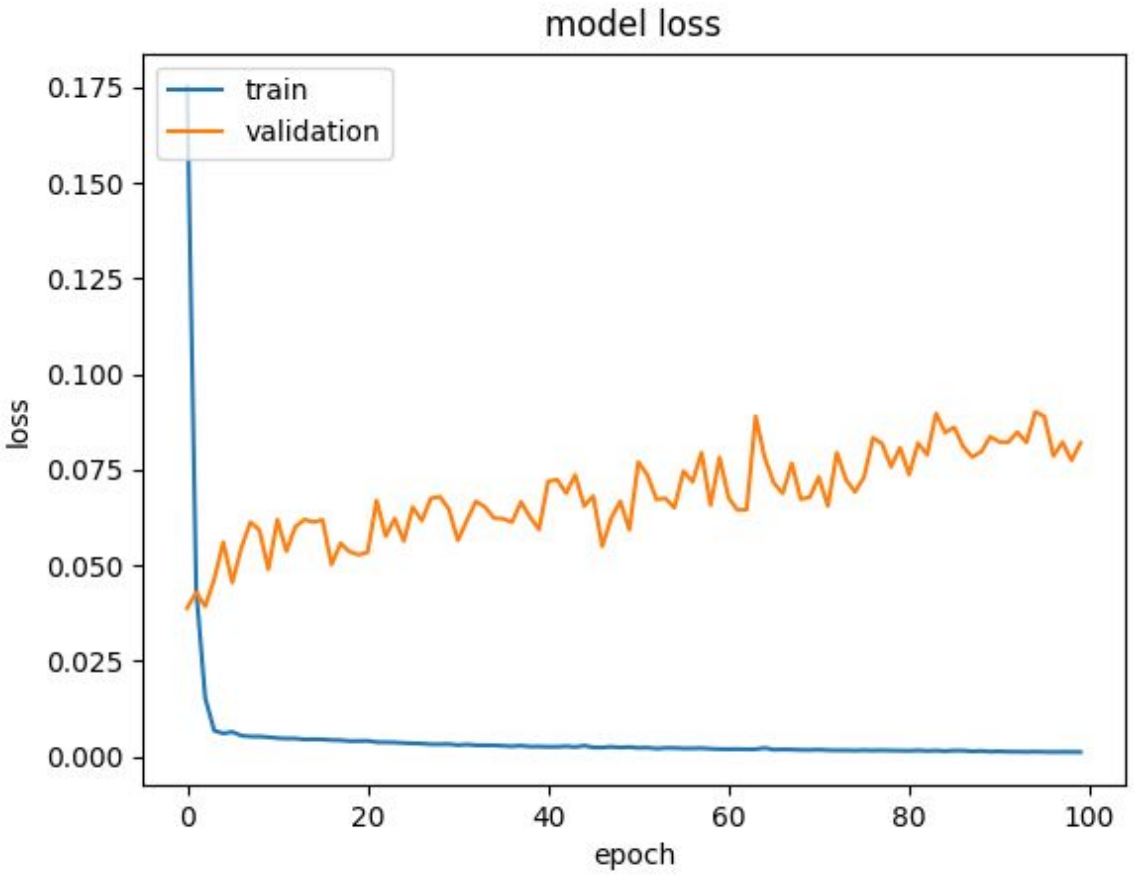


Delay Loss

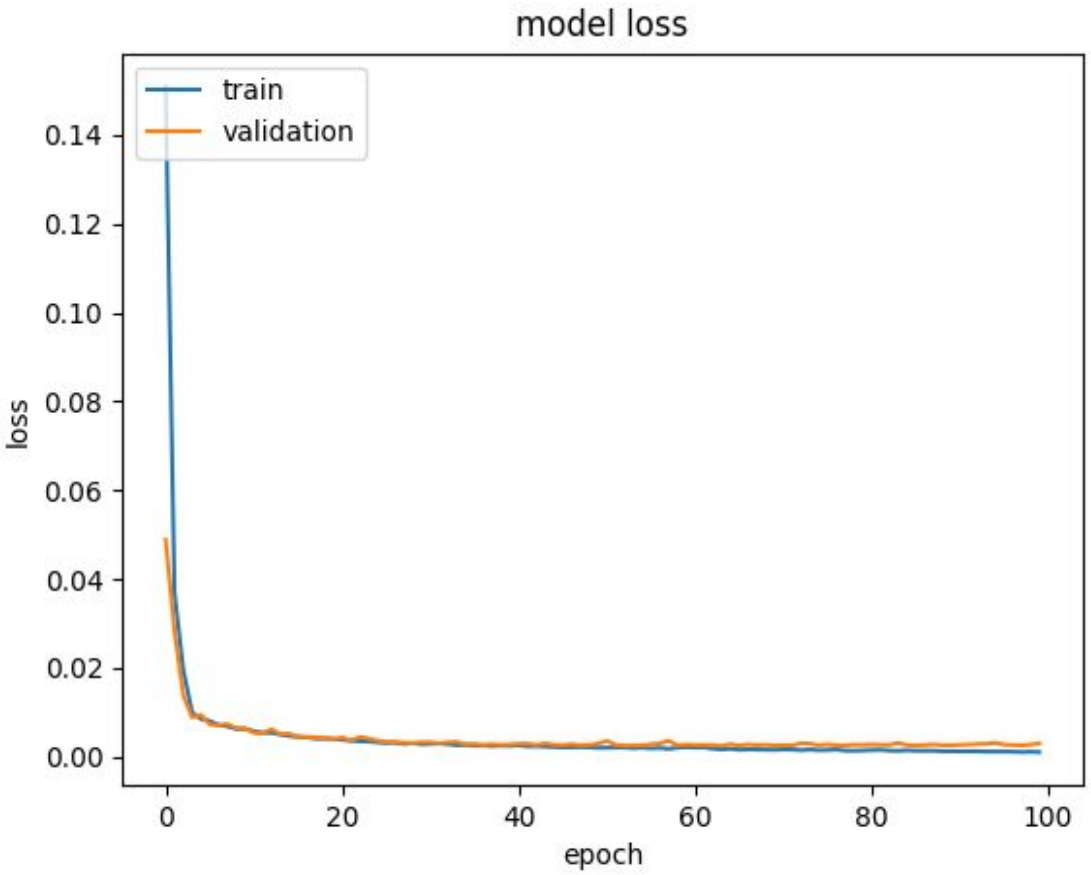
Resultados obtidos

- Análise do modelo

Divisão por Cenários:



Divisão por Topologias:



Pdr Loss

Resultados obtidos

- Comparação com o Ns-3

Model Trained By Division		Scenarios			Topologies			All Data	
	Throughput	Delay	Pdr	Throughput	Delay	Pdr	Throughput	Delay	Pdr
(%)	49	66	25	20	48	12	12		34

Conclusões

No fim deste trabalho conseguimos tirar algumas conclusões importantes para o futuro aprofundamento deste tema:

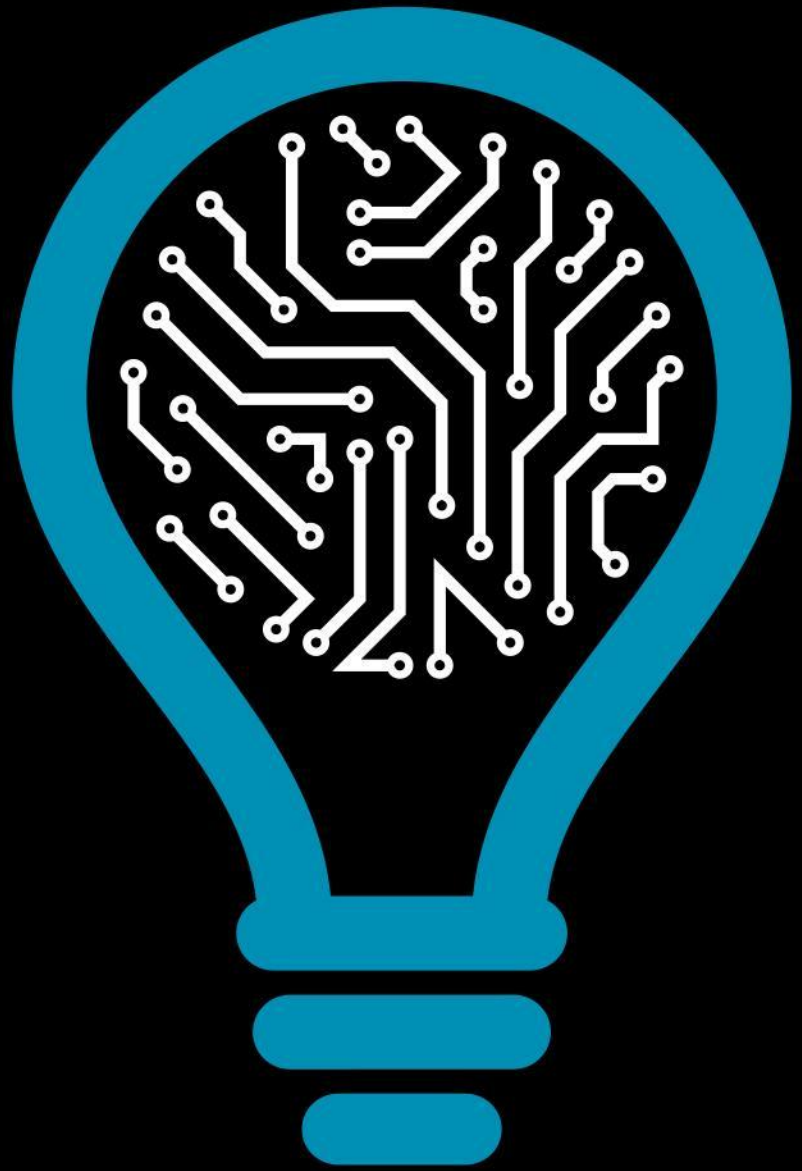
- o modelo dá bons resultados, mas necessita de mais dados, principalmente cenários, de forma a dar resultados mais próximos da realidade;
- o número de *epochs* de treino, no caso do delay, deveria ser aumentado. Acreditamos que com mais de 100 *epochs* o delay melhora substancialmente;
- o modelo dá melhores resultados quando treinado com *distance encoding*;
- a divisão por topologias nos dados gera um modelo com melhores resultados;

Feedback

- O que correu bem?

- O que correu menos bem?

Questões?



INESCTEC

INSTITUTO DE ENGENHARIA
DE SISTEMAS E COMPUTADORES,
TECNOLOGIA E CIÊNCIA