

# Processing node prevision for network services

---

# Contextualização do problema

Objetivos:

- Reduzir reencaminhamentos pela rede
- Selecionar os melhores nodes para processamento de serviços
- Utilização de modelos de previsão



# Dataset

Request					Network State												Labels							
id	src	dst	bw	duration	S0	S1	S2	S3	S4	MLU	MNU	E0	E1	...	E26	E27	N0	...	N10	RS0	RS1	RS2	RS3	RS4

Ordered list of requested service functions

Identify at which node the requested service  $S_i$  will be processed.

id :

src:

dst:

bw:

duration:

Service request ID

Request source node

Request destination node

Requested bandwidth

Time needed to process the request

MLU:

MNU:

$E_i$ :

$N_i$ :

Maximum link utilization

Maximum Node utilization

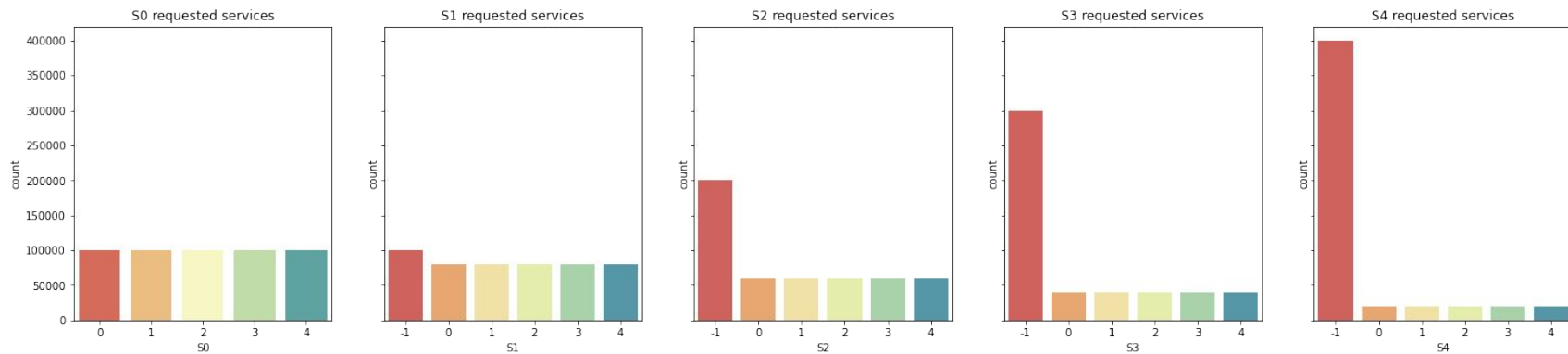
Usage of link  $E_i$

Usage of node  $N_i$

## Request of Service Function Chaining:

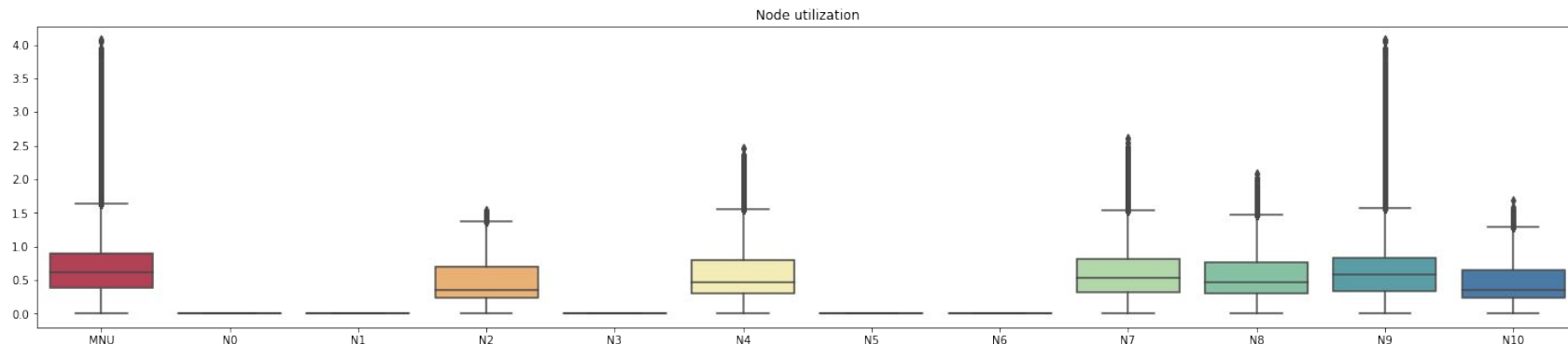
Src	Dst	Services	Duration	Bandwidth
2	7	[0,3,2]	10	5000

# Análise exploratória do dataset - Serviços



Distribuição equilibrada de serviços pedidos pelo dataset. Contribui para resultados representativos, dados de testes são proporcionais a dados de treino.

# Análise exploratória do dataset - Nodes



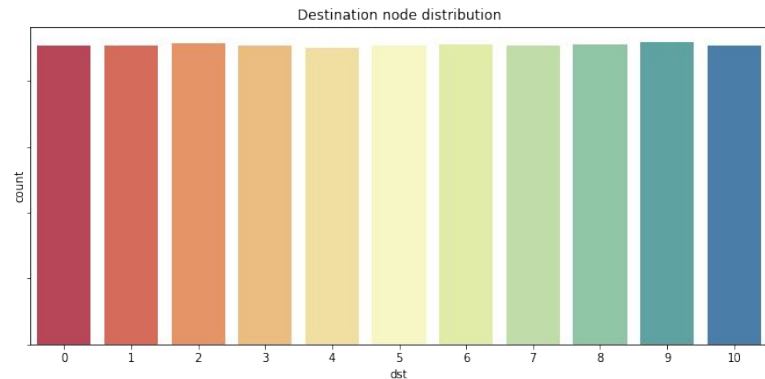
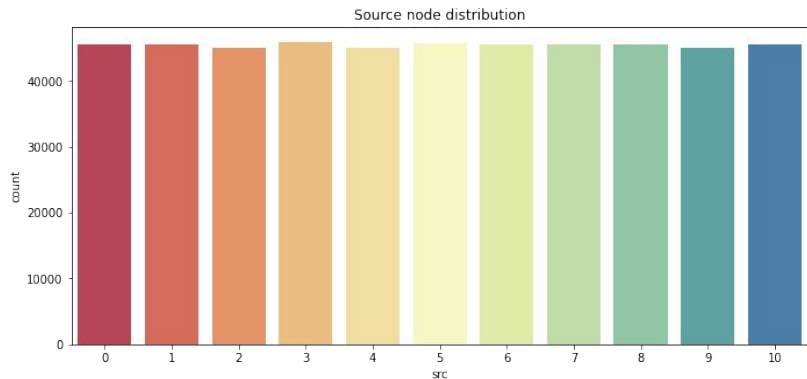
## Node Services

2	[0, 1]
4	[2, 3]
7	[2, 4]
8	[0, 4]
9	[0, 1, 3, 4]
10	[1]

## Service Functions:

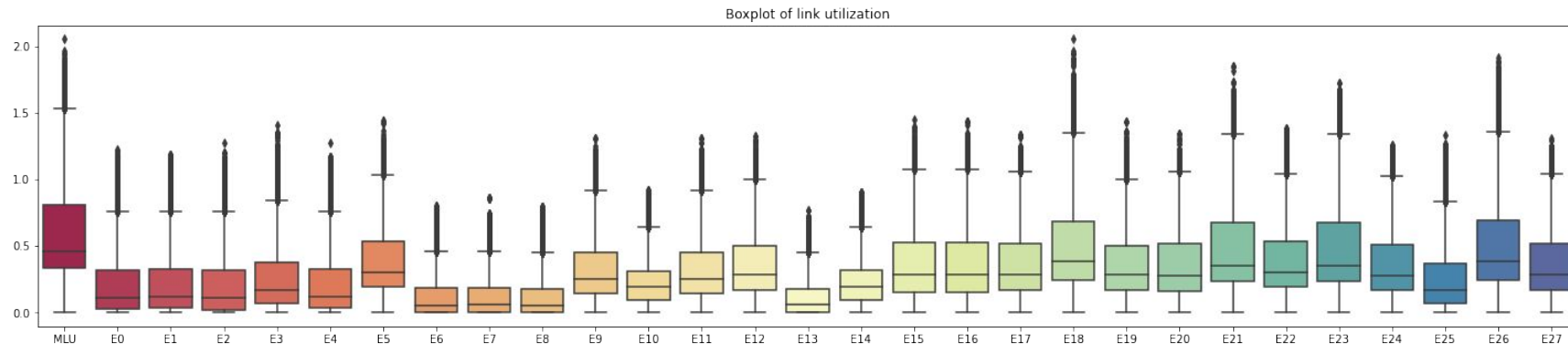
0	Firewall,
1	WAN and application acceleration,
2	Deep Packet Inspection (DPI),
3	Lawful Intercept (LI),
4	server load balancing,

# Análise exploratória do dataset - Nodes



Embora existam nodes que não prestam serviços na rede, encontra-se uma distribuição equilibrada de todos os nodes, tanto como nodes de origem como de destino.

# Análise exploratória do dataset - Links



Posteriormente, foi verificado que estes dados não contribuem tanto para a previsão como a utilização dos nodes.

# Processamento do dataset

- Eliminação das colunas relativas a id, MNU, MLU e nodes sem carga
- Normalização dos valores de *bandwidth*
- One hot encoding de variáveis categóricas (src, dst, S[0-4])

Posteriormente, foram testadas variações eliminando colunas, obtendo melhores resultados com a eliminação das seguintes colunas:

- Utilização dos links (E[0-27])
- Duração dos pedidos



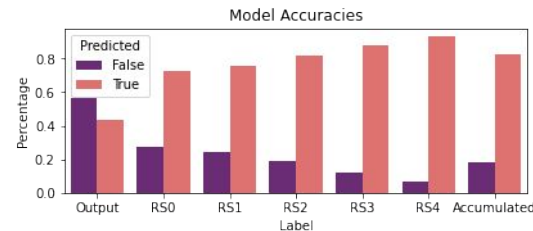
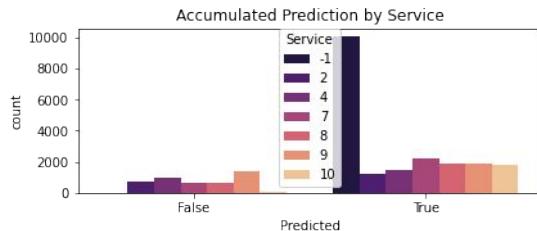
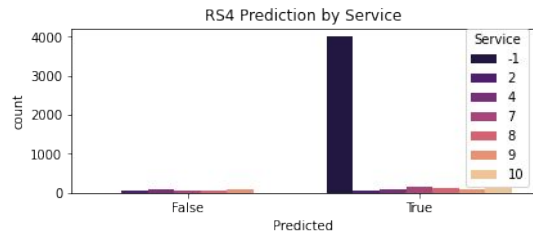
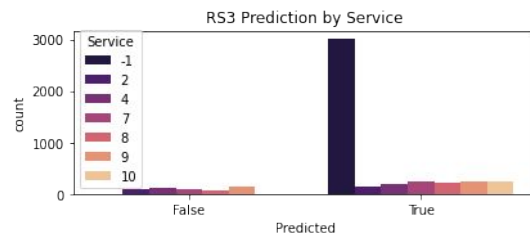
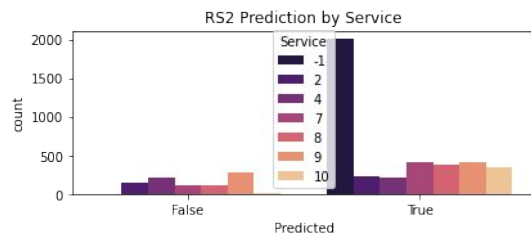
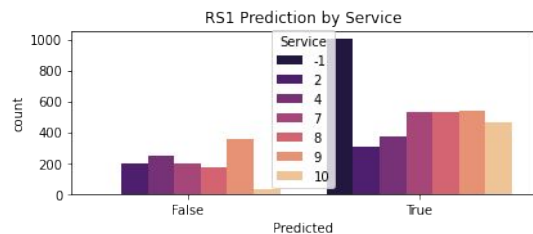
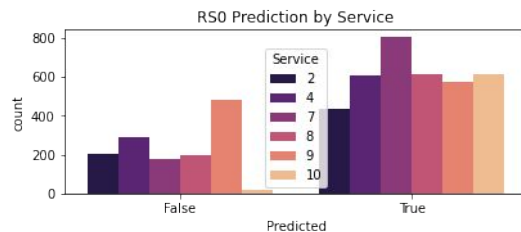
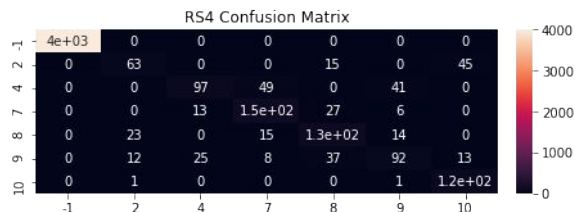
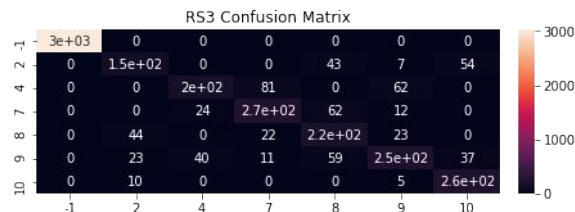
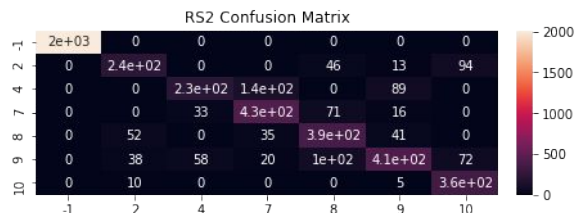
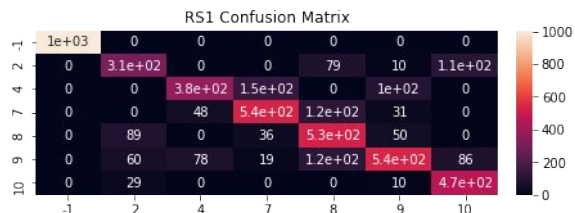
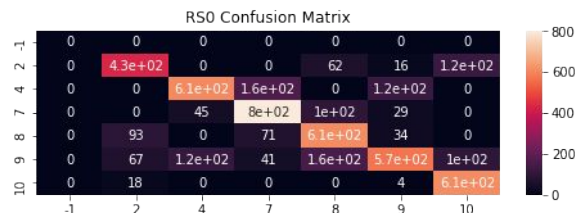
# Previsão com modelos de multi-output

Numa primeira fase, foram aplicados ao dataset modelos que fizessem a previsão dos 5 nodes (RS[0-4]) para os 5 serviços pedidos (S[0-4]). Para tal, foram necessários modelos que suportam multi-output. Foram testados os seguintes modelos:

- KNeighborsClassifier
- RandomForestClassifier
- RadiusNeighborsClassifier
- DecisionTreeClassifier
- ExtraTreeClassifier

Devido à dimensão do dataset e a limitações de memória no Google Colab, estes modelos foram treinados sobre um dataset reduzido, com 100,000 observações e 5% para validação.

# RandomForestClassifier, accuracy 0.4346



## Previsão com modelos single-output

De forma a melhorar resultados, foi feita uma transformação do dataset para a previsão de um serviço de cada vez. Para cada entrada do dataset anterior, são geradas até 5 novas entradas, tendo em conta os valores de  $S[0-4]$  e  $RS[0-4]$ . Mantém-se as variáveis de estado de rede, e introduz-se um serviço e node para cada entrada.

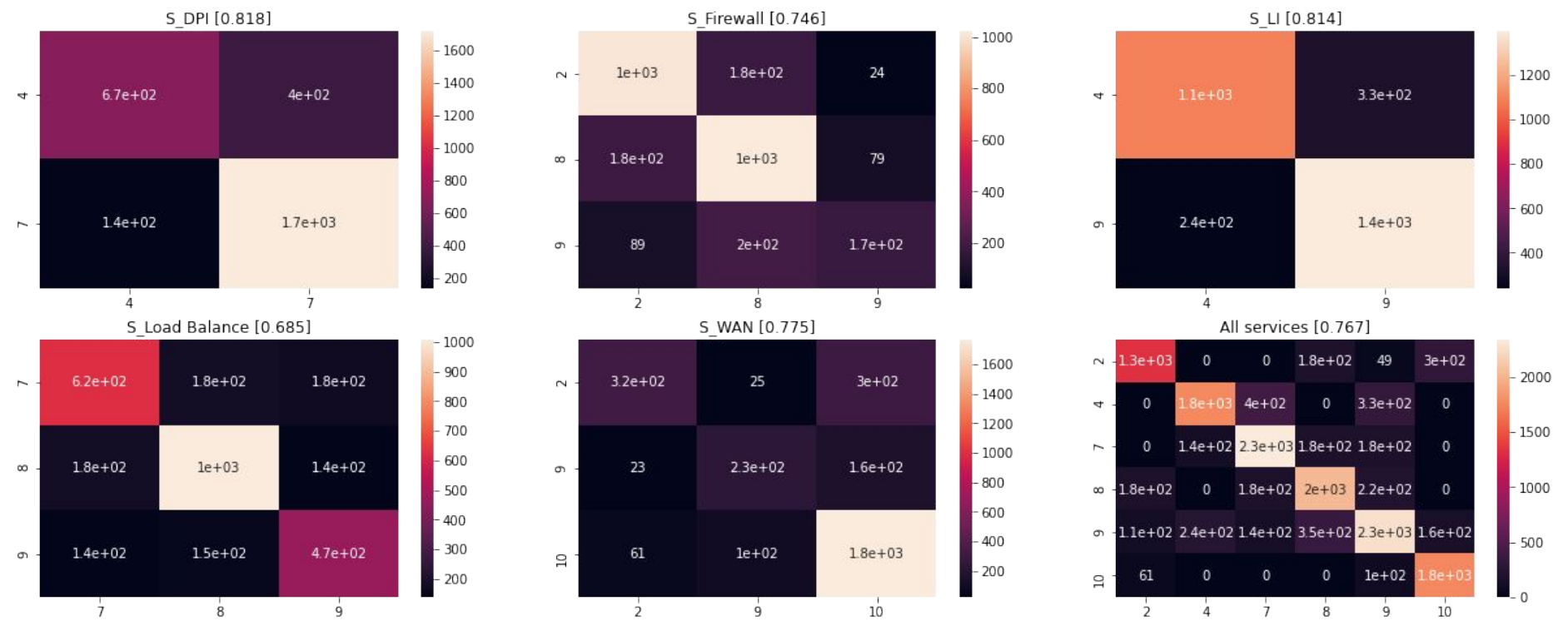
Além disso, foi treinado um modelo para cada tipo de serviço. Assim, quando é feita a previsão, cada serviço é aplicado no seu modelo específico.

# Previsão com modelos single-output

Após o processamento para obter o novo dataset orientado a cada serviço, obtivemos 1,500,000 entradas, pelo que, mais uma vez, os modelos foram treinados com apenas uma fração destes dados, proporcional ao testado anteriormente (300,000 observações com 5% para validação). Foram testados os seguintes modelos:

- SVC (com kernel linear)
- RandomForestClassifier
- KNeighborsClassifier
- SGDClassifier
- DecisionTreeClassifier

# RandomForestClassifier, accuracy 0.767



# Abordagem inicial em Deep Learning

Foram implementados redes básicas Feedforward, recorrendo a camadas Dense e Dropout, com uma Dense layer final com ativação *softmax*, para classificação. No entanto, os resultados não são muito promissores, sugerindo overfitting. Foi aplicada a abordagem da classificação serviço a serviço, sendo desenvolvido uma rede para classificação de cada tipo destes.

Será necessário desenvolvimento adicional para obter resultados superiores aos modelos de previsão descritos anteriormente. Uma abordagem potencialmente superior será o desenvolvimento de redes recorrentes.

# Planeamento

- Desenvolvimento adicional da Feedforward Neural Network
- Implementação de Recurrent Neural Networks (possivelmente LSTM)
- Otimização dos modelos abordados
- Revisões finais e análise de resultados