NPSmart Tool

Qualidade de dados

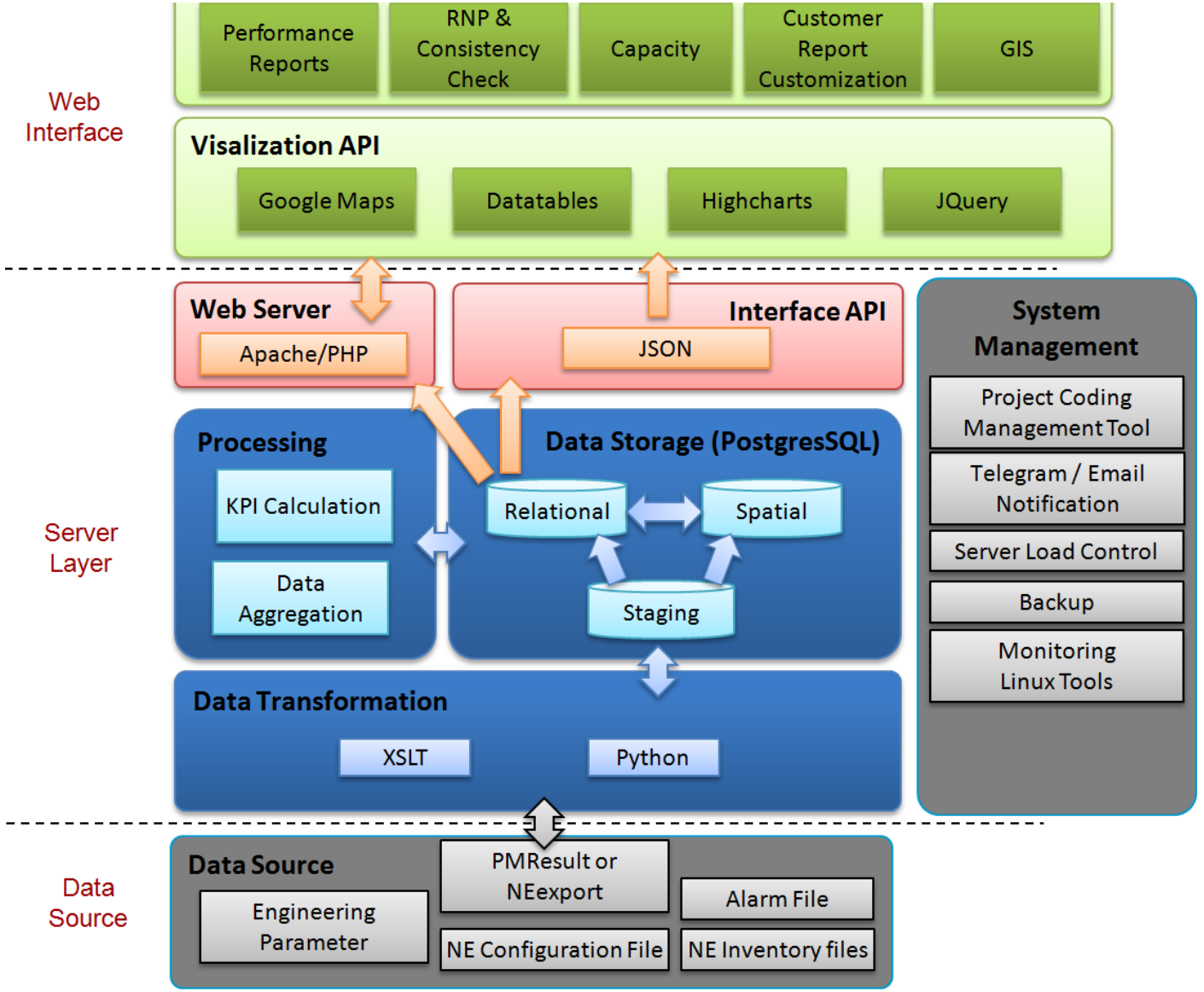
09/2017

Carlos Eduardo Covas Costa

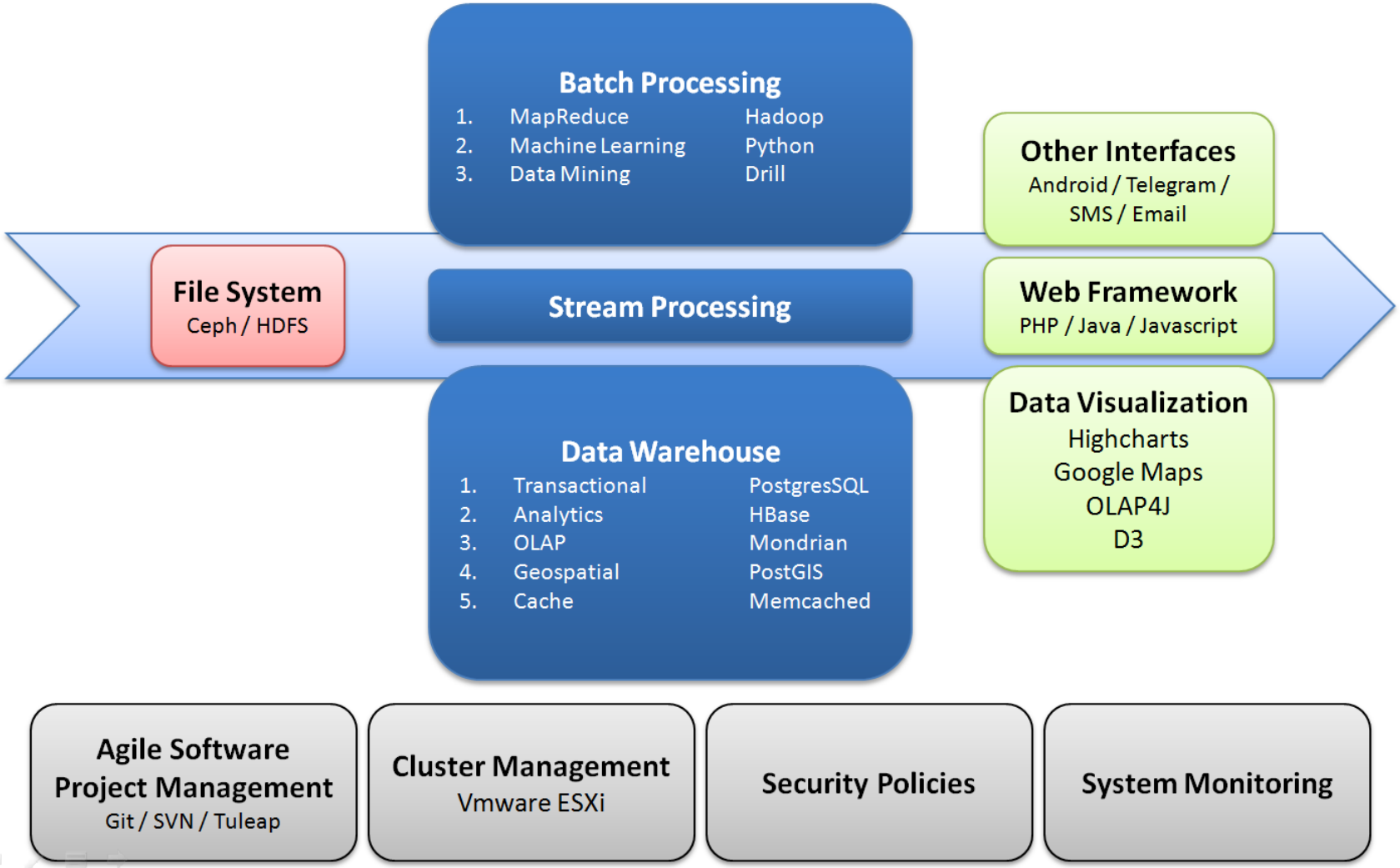
Contents

Collecting	Database	Intelligence	Interface
Tool for automatic data Collecting Easy adaptaion for G/U/L	Database customized for Huawei Engineering / Easy data-mining, analytics	Procedures Scripts calculate KPIs and correlate Information; Business Model; Machine Learning	Rich user interface focusing on Presentation and easy use / Customization

NPSmart tool - Designed Architecture



NPSmart tool - Prospect Technologies



Motivação

- Dificuldade maior para realização de tarefas de monitoramento e detecção de falhas.
- Necessidade de bases de dados confiáveis, com dados consistentes e sem dados faltantes.
    - Sistema de monitoramento de qualidade de dados
- Aproveitar um sistema de monitoramento de qualidade dos dados para apresentar alertas de degradações.

Objetivo

- Utilizar Redes neurais para modelagem de séries temporais no contexto de indicadores de redes móveis.
- Comparar o desempenho da rede neural com o método estatísticos de análise de séries temporais ARIMA.
- Propor corredores de validação que apresentem níveis de credibilidade da amostra.
- Utilizar tais corredores para detectar intrusos, bem como gerar alertas de possíveis degradações na rede.

Introdução

• O processo de ETL, principalmente nas partes aquisição de dados, transformação de dados, cálculo de KPIs e agregação de KPIs, gera erros que comprometem a qualidade de dados.
E.g.

- dados faltantes oriundos de falha na aquisição dos dados ou erros na transformação devido a mudanças no formato.
- agregações com menos amostras produzem valores errados por só considerarem parte da infomação

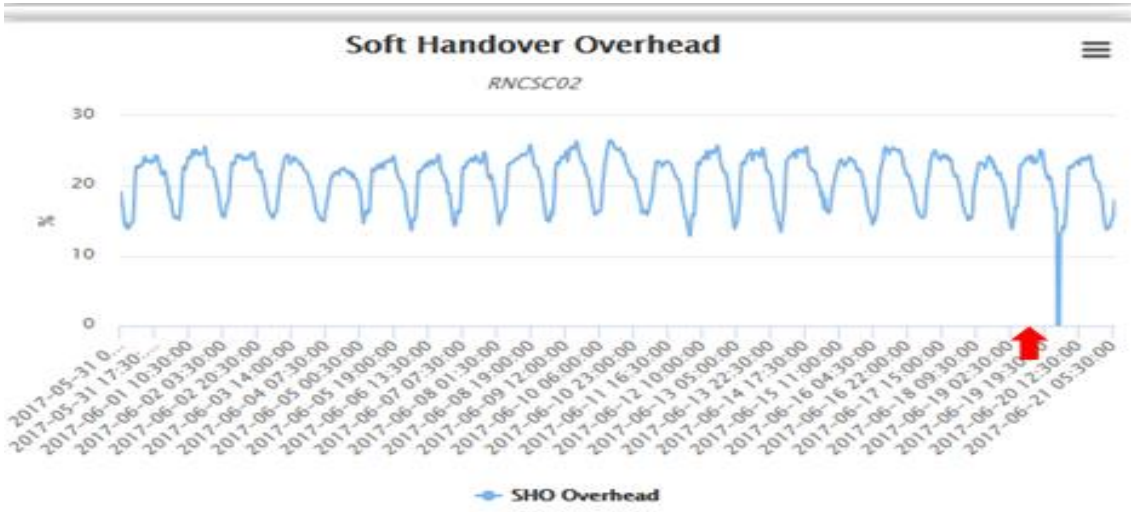
• Portanto, faz-se necessário a utilização de técnicas de modelagem de sistemas dinâmicos afim de introduzir um processo de monitoramento da qualidade de dados, que seja capaz de

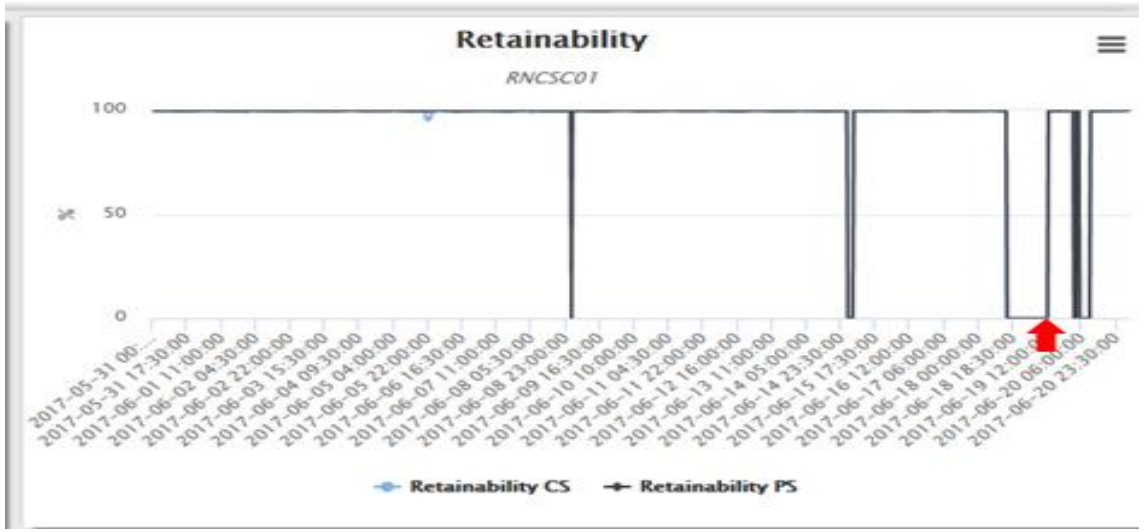
- substituir amostras faltantes pelo valor previsto.

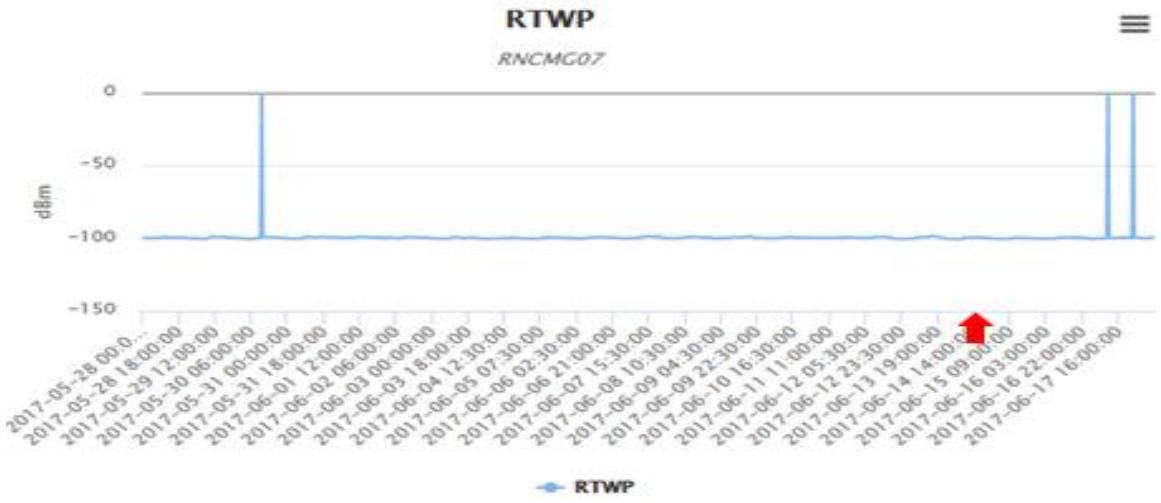
- detectar anomalias, infomando-as a um engenheiro para análise.

Introdução

Exemplos de gráficos retirados de bases com qualidade de dados ruim.

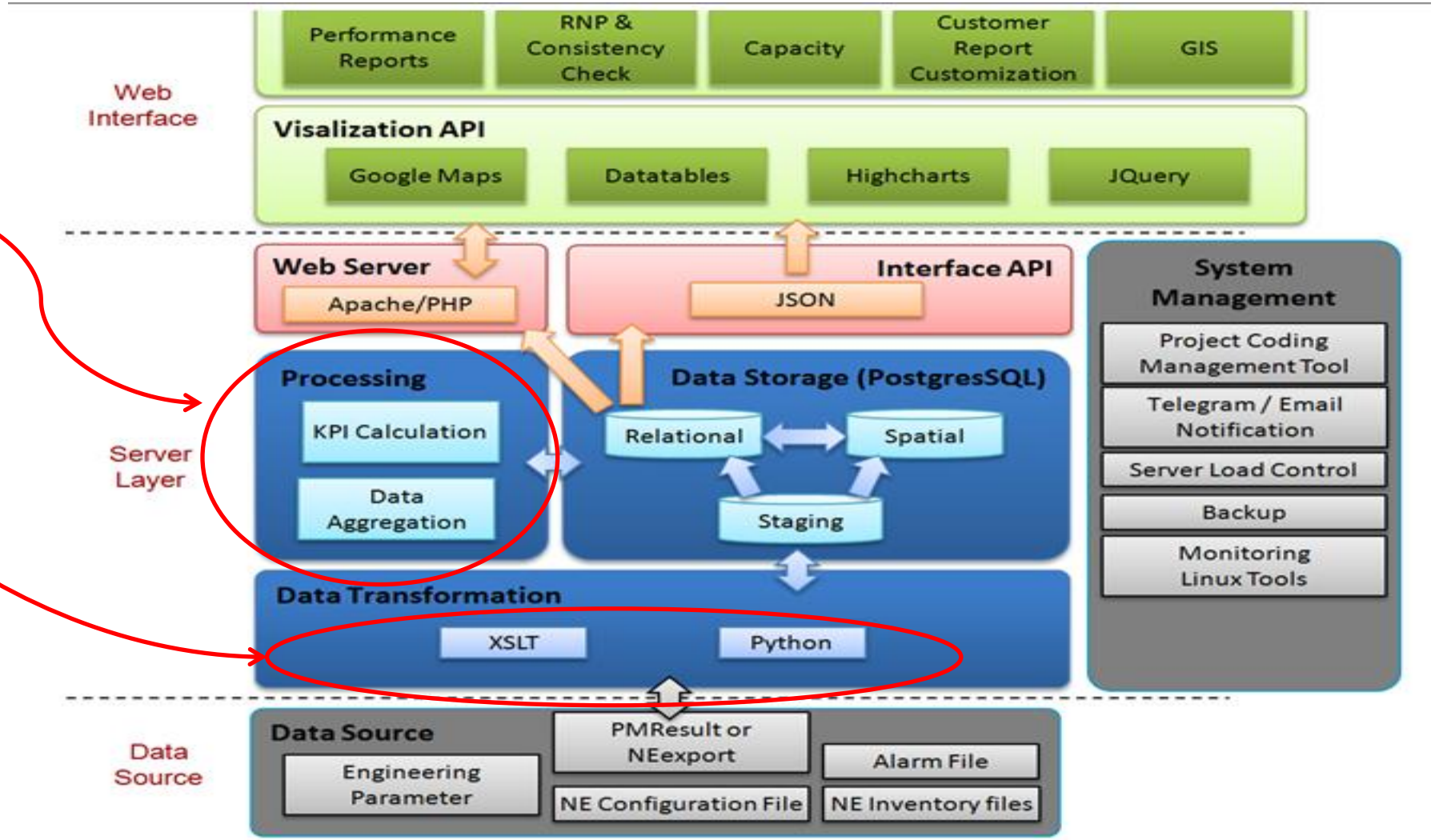






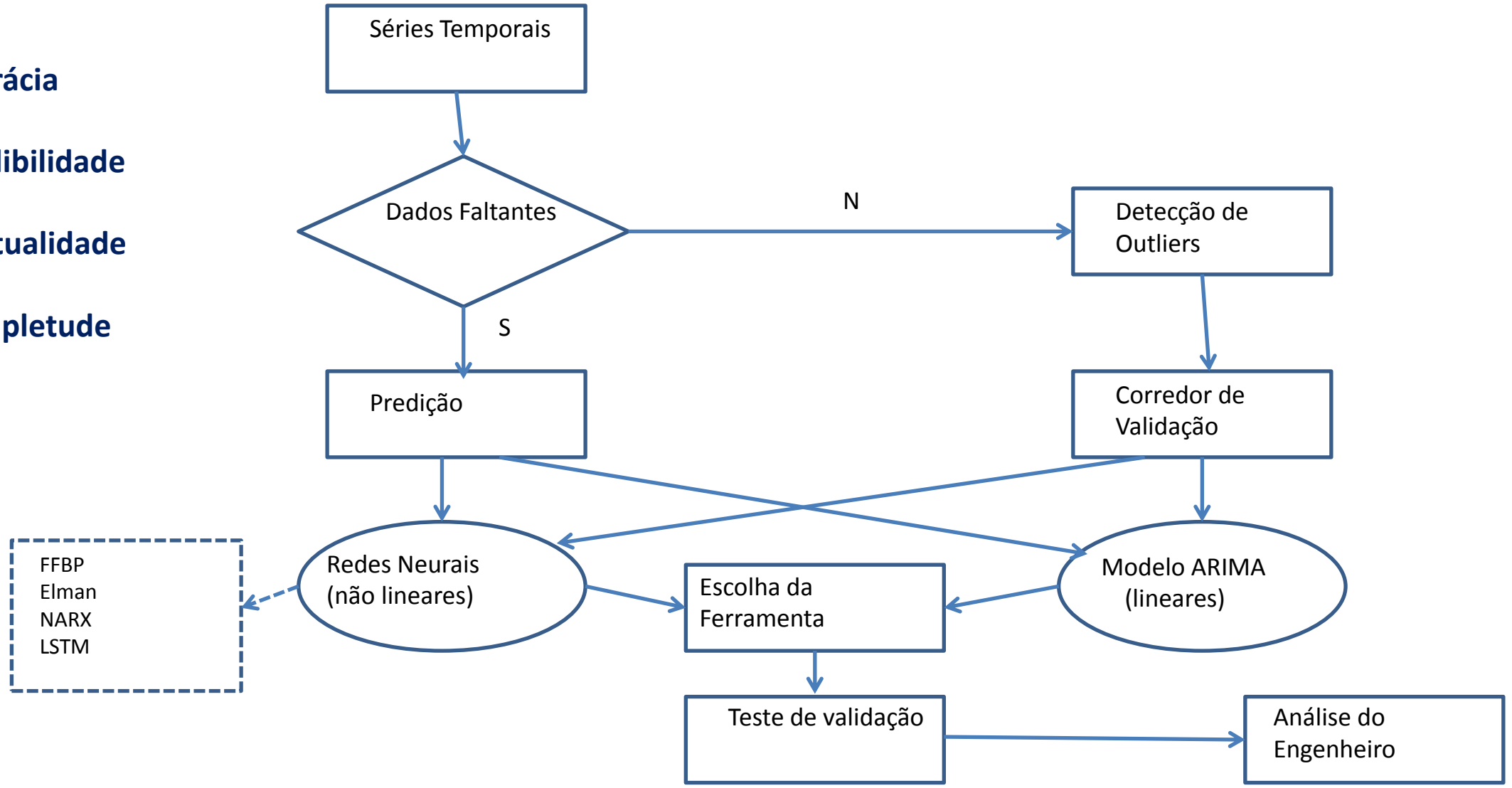
Introdução

Introdução de erros na base



Introdução – Qualidade de dados em Séries temporais[4]

- **Acurácia**
- **Credibilidade**
- **Pontualidade**
- **Completude**



Introdução - Análise de ST

•Normalmente, os procedimentos de análise estatística de séries temporais supõe que estas sejam estacionárias.

• Desta forma, é necessário verificar a estacionariedade das séries temporais que se deseja modelar, assim como definir um procedimento para tornar estacionárias as que não o sejam previamente.

• Podemos considerar que cada série temporal é uma realização de um processo estocástico, afim de estender, para estas, o conceito de estacionariedade.

Definição: Um processo estocástico Z = {z(t), t pertencente T} é dito estacionário quando seus momentos estatísticos não variam no tempo[4]:

$$i \quad E\{z(t)\} = \mu(t) = \mu$$

$$ii \quad Var\{z(t)\} = \sigma^2(t) = \sigma^2$$

$$iii \quad \gamma(t_1, t_2) = Cov\{z(t_1), z(t_2)\} \text{ é função somente de } |t_1 - t_2|$$

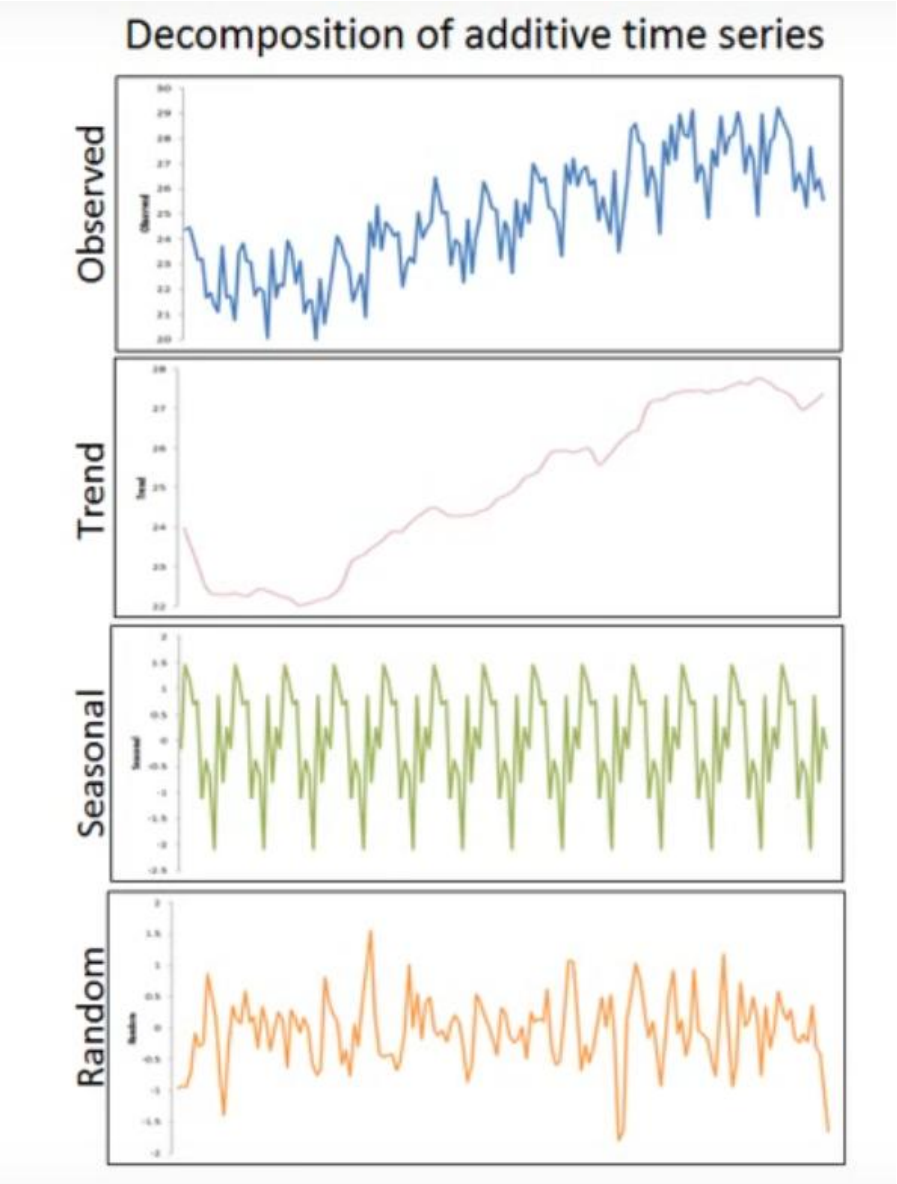
Introdução - Análise de ST

• **Tendência** – comportamento de longo prazo.
  •Ex.: crescimento populational mostra uma tendência de aumento.

• **Ciclos –** flutuações nos valores da variável que se repetem com certa periodicidade.
  •Variações cíclicas de da economia como prosperidade, declínio, depressão e recuperação.

• **Sazonalidade** – Comportamento sistemático e relacionado ao calendário. Pode ser identificado por intervalos regularmente espaçados de picos ou vales que ocorrem com a mesma magnitude, aproximadamente todo ano.
  • Ex.: acréscimo nas vendas de varejo em decorrência do natal

• **Componente irregular –** Variações aleatórias causadas por fatores imprevisíveis.



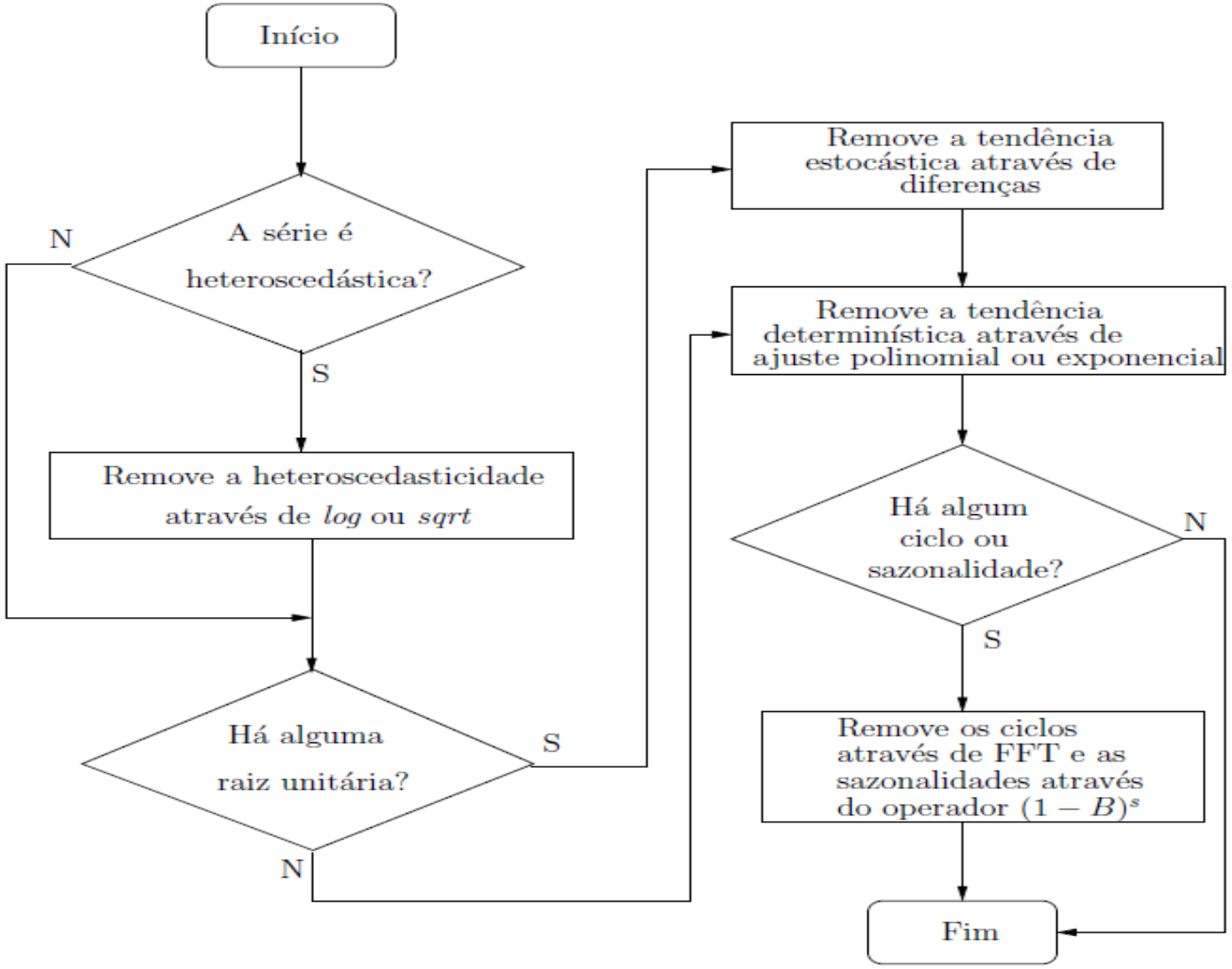
Metodologia

- Neste trabalho a seguinte metodogia foi adotada para análise, tratamento e modelagem da séries temporais usadas.

    - Análise visual do comportamento da série, afim de notar variação dos momentos estatísticos.
    - Realização de teste ADF (Augmented Dickey-Fuller) para verificação de probabilidade de existência de raiz unitária.
    - Estimação da tendência determínistica
    - Utilização de FFT para detecção de sazonalidade.
    - Decomposição da série em Tendência, sazonalidade, ciclos senoidais e resíduo.
    - Visualização das funções de autocorrelação e correlação parcial.
    - Determinação da medida AIC para determinação do modelo ARIMA.
    - Teste de diferentes topologias de redes neurais perceptron de múltiplas camadas com uma camada intermediária.
    - Análise do histograma do erro
    - Determinação de corredores de validação para detecção de outliers.

Todo o trabalho foi feito utilizando a linguagem Python, principalmente as bibliotecas Numpy, Scipy, Pandas, Matplotlib e Keras.

Metodologia – Procedimento para estacionarização [4]



Metodologia

O modelo ARIMA de Box & Jenkins é um dos métodos mais utilizados na análise de séries temporais. Este possui restrições para séries lineares e estacionárias e assume que a série siga uma distribuição estatística conhecida (e.g. distribuição normal). Ele possui, como casos particulares, os métodos Autoregressivo (AR) e Médias móveis (MA), sendo o valor futuro de uma série univariada uma função linear de observações passadas e erros aleatórios.

No processo estocástico AR(p), o valor futuro de uma variável assumido como sendo uma combinação linear de p observações passadas e um erro aleatório juntamente com um termo constante

$$x(t) = C + \sum_{i=1}^{p} \alpha_i * x_{t-1} + \epsilon_t$$

onde alphai, alphap são os parâmetros e p a ordem do modelo, C uma constante e o erro aleatório e epsilon(t) um ruído branco.

No processo MA(q) os erros de previsões passados da equação de previsão são usados como a variável explicativa conforme fórmula abaixo:

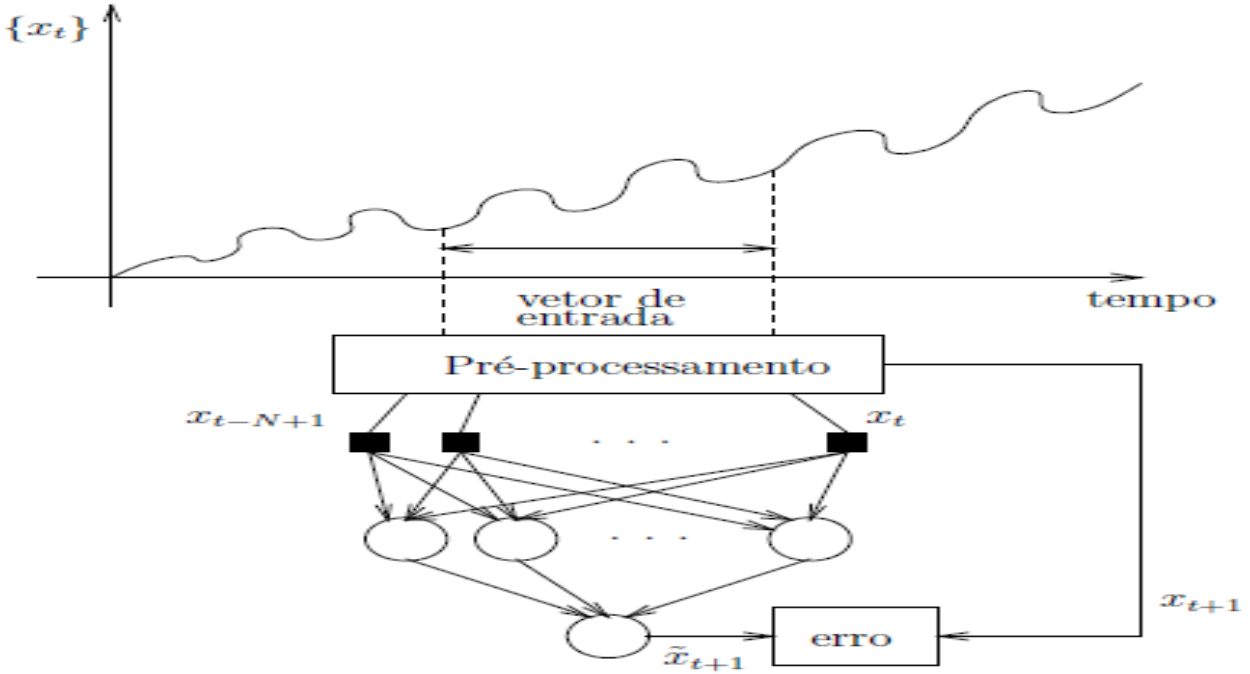
onde theta i, thetaq sçao os parâmetros do modelo, q a ordem, a média das séries e epsilon(t), epsilont(t-1) os erros aleatórios.

$$x(t) = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i * \epsilon_{t-1}$$

Metodologia

A arquitetura mais tradicional no uso de RNAs na modelagem de séries temporais é a estrutura que propaga o sinal de entrada (feedforward, FF) através de camadas de neurônios e retropropaga o erro de saída (backpropagation, BP) para proceder o ajuste de pesos sinápticos.

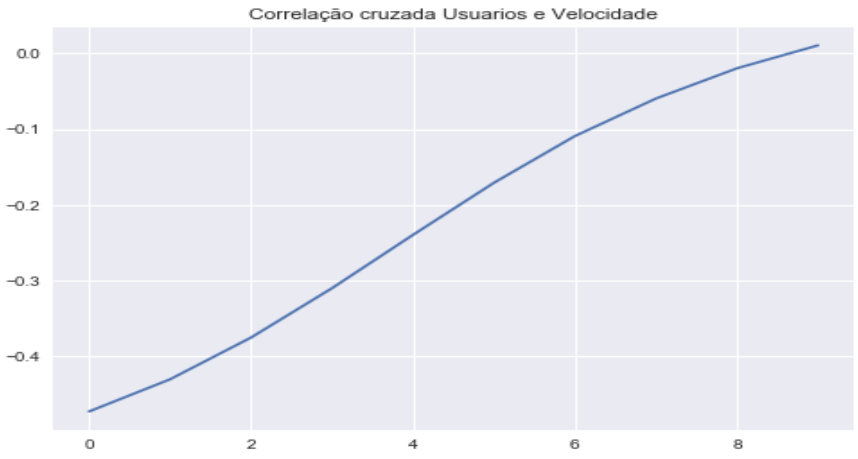
O modelo Perceptron de múltiplas camadas é caracterizado por uma rede de duas ou mais camadas, sendo uma de entrada (input layer), uma ou mais camadas escondidas (hidden layer) e uma saída (output layer), conectada por links. Em previsão de séries, os dados de entrada consistem nos atrasos de tempo, ou os valores da séries nos instantes anteriores. [4]

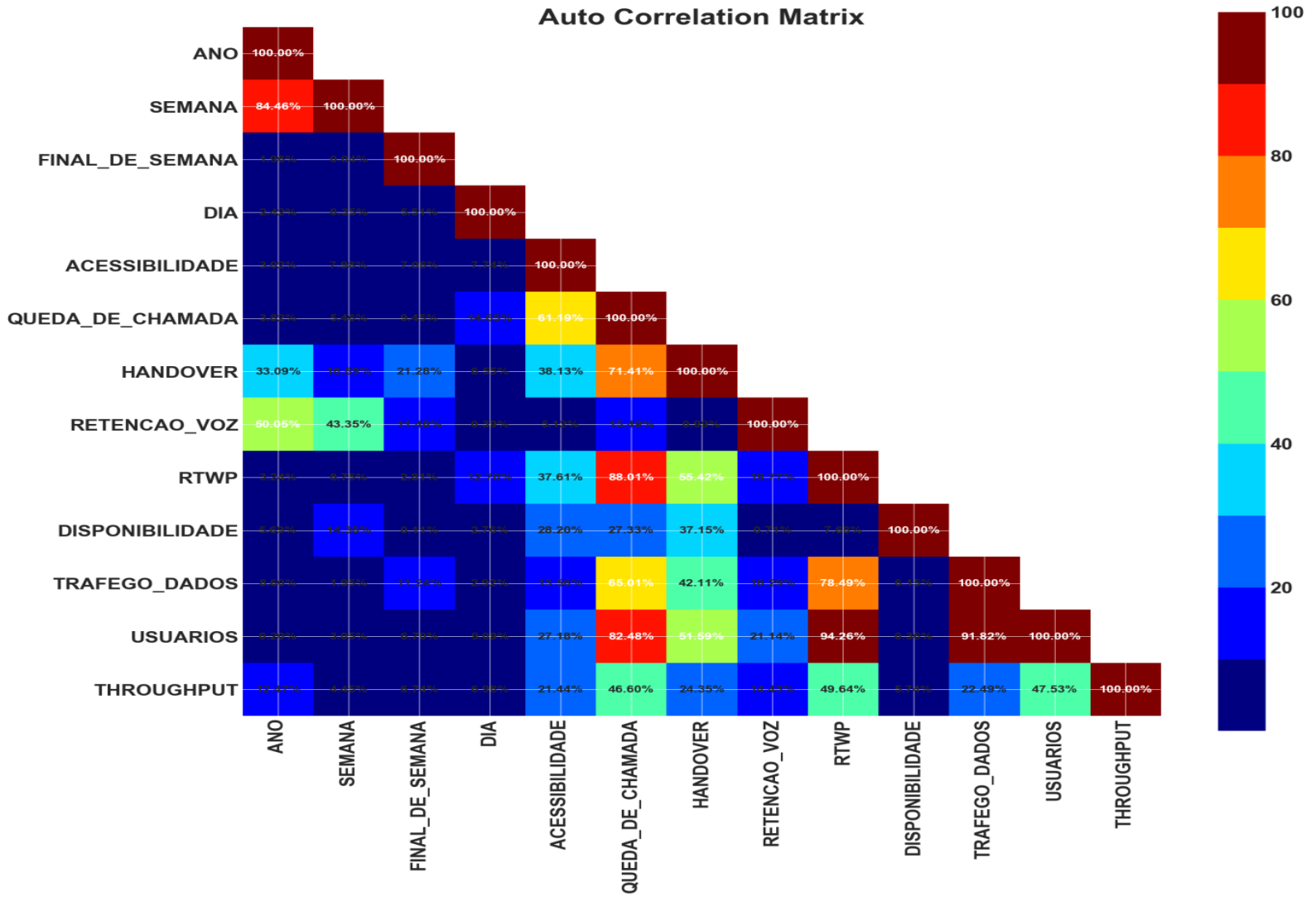


Dataset

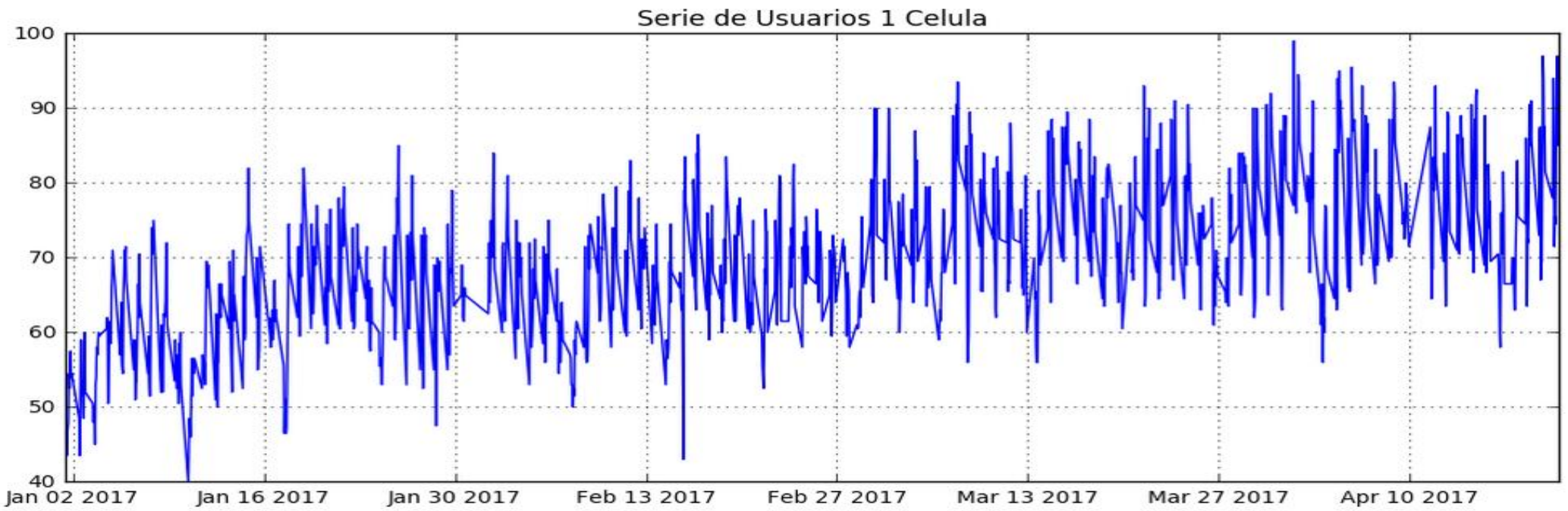
Conjunto de dados é composto por
1 célula de rede escolhida por
apresentar comportamento crescente durante
os meses e refere-se ao período de 1 de janeiro
a 20 de abril.

Conjunto: 1278
Treinamento: 851 registros
Validação: 427

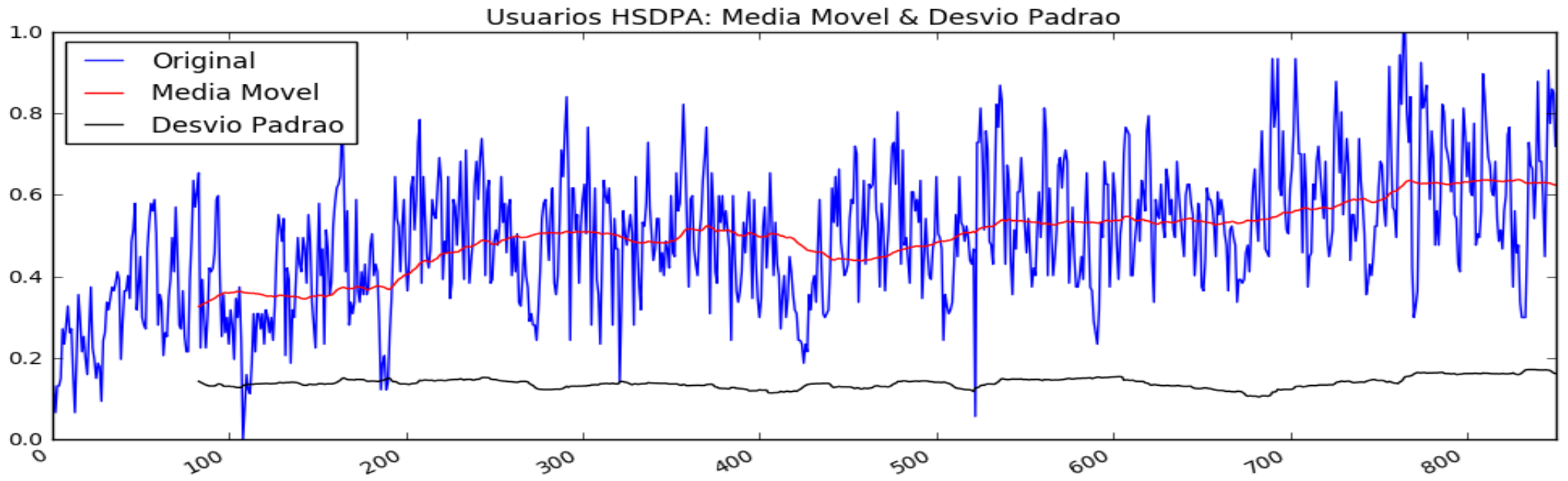




Dataset


Serie de Usuarios 1 Celula
100
90
80
70
60
50
40
Jan 02 2017
Jan 16 2017
Jan 30 2017
Feb 13 2017
Feb 27 2017
Mar 13 2017
Mar 27 2017
Apr 10 2017

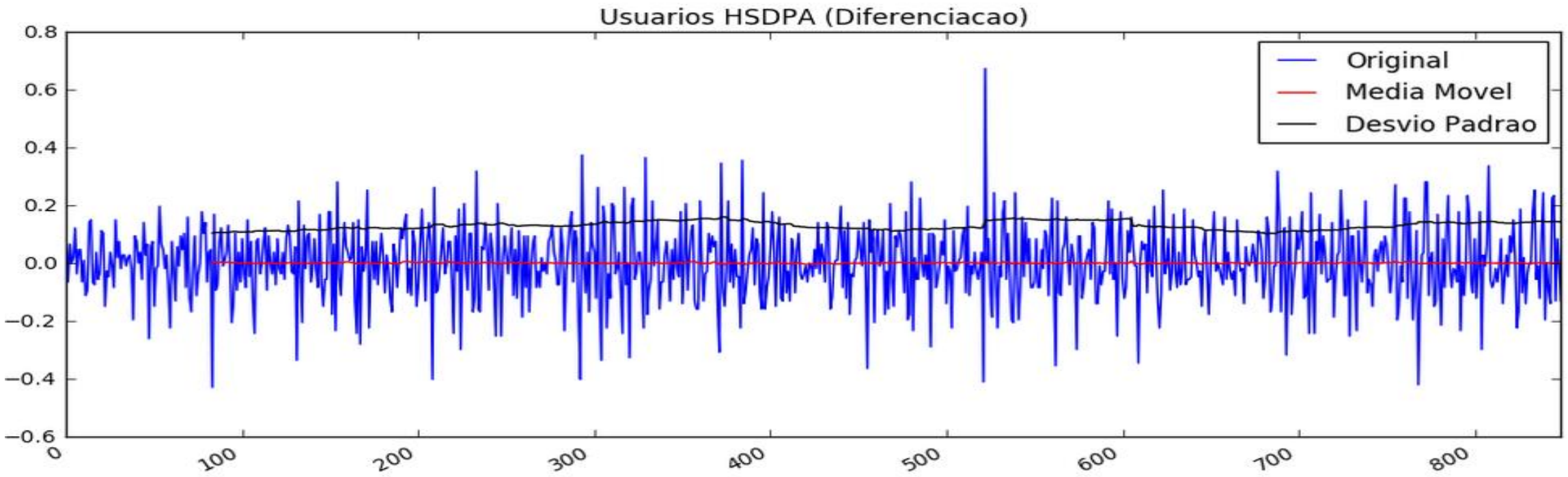
ARIMA - Teste de estacionariedade




```
Resultados do teste Dickey-Fuller:
Teste estatistico          -4.135775
p-value                     0.000845
#Lags Usado                17.000000
# Observacoes usado       833.000000
Valor Critico (5%)         -2.865016
Valor Critico (1%)         -3.438225
Valor Critico (10%)        -2.568621
dtype: float64
```

→ Hipótese de Raiz Unitária Rejeitada → Tendência Determinística

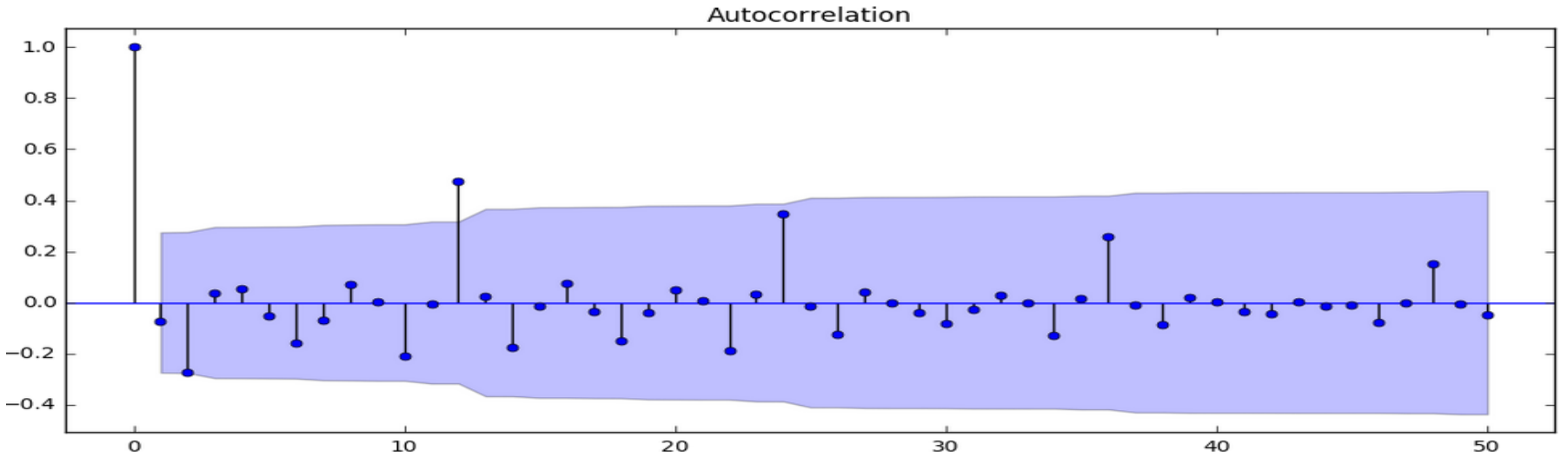
ARIMA - Diferenciação

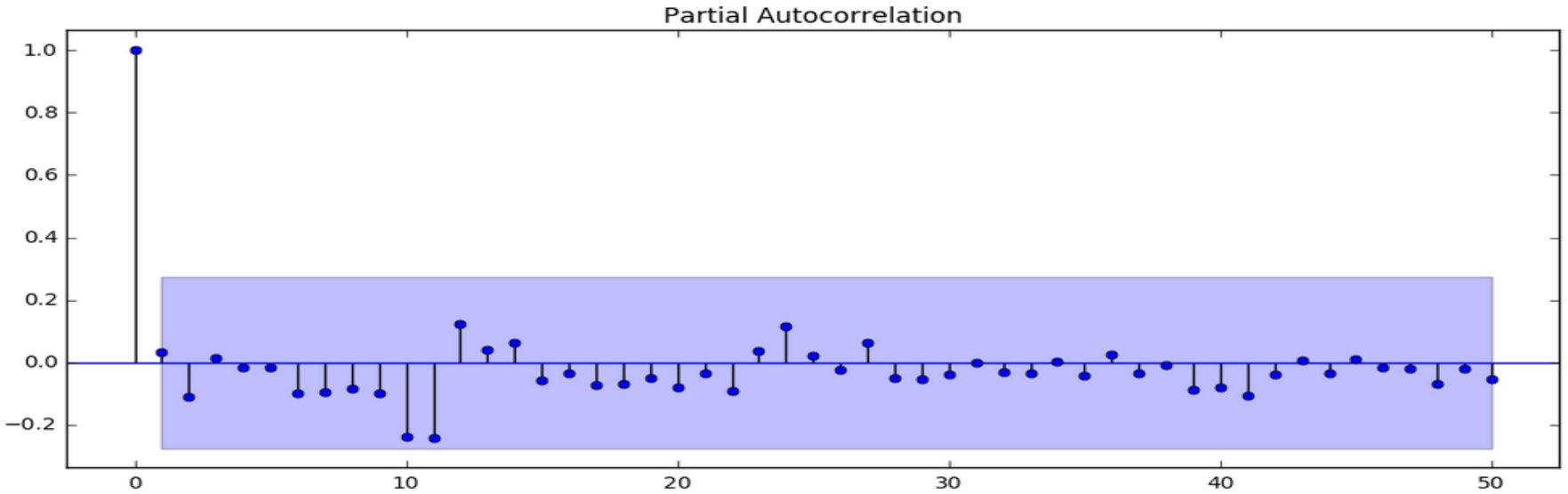


Filtro Passa-Alta

$$\mathbf{M} = \begin{pmatrix} & & & \vdots & & & \\ \dots 1 & -1 & 0 & 0 & 0 & 0 & 0 \dots \\ \dots 0 & 1 & -1 & 0 & 0 & 0 & 0 \dots \\ \dots 0 & 0 & 1 & -1 & 0 & 0 & 0 \dots \\ \dots 0 & 0 & 0 & 1 & -1 & 0 & 0 \dots \\ \dots 0 & 0 & 0 & 0 & 1 & -1 & 0 \dots \\ \dots 0 & 0 & 0 & 0 & 0 & 1 & 1 \dots \\ & & & \vdots & & & \end{pmatrix}$$

ARIMA - Correlograma





ARIMA - AIC

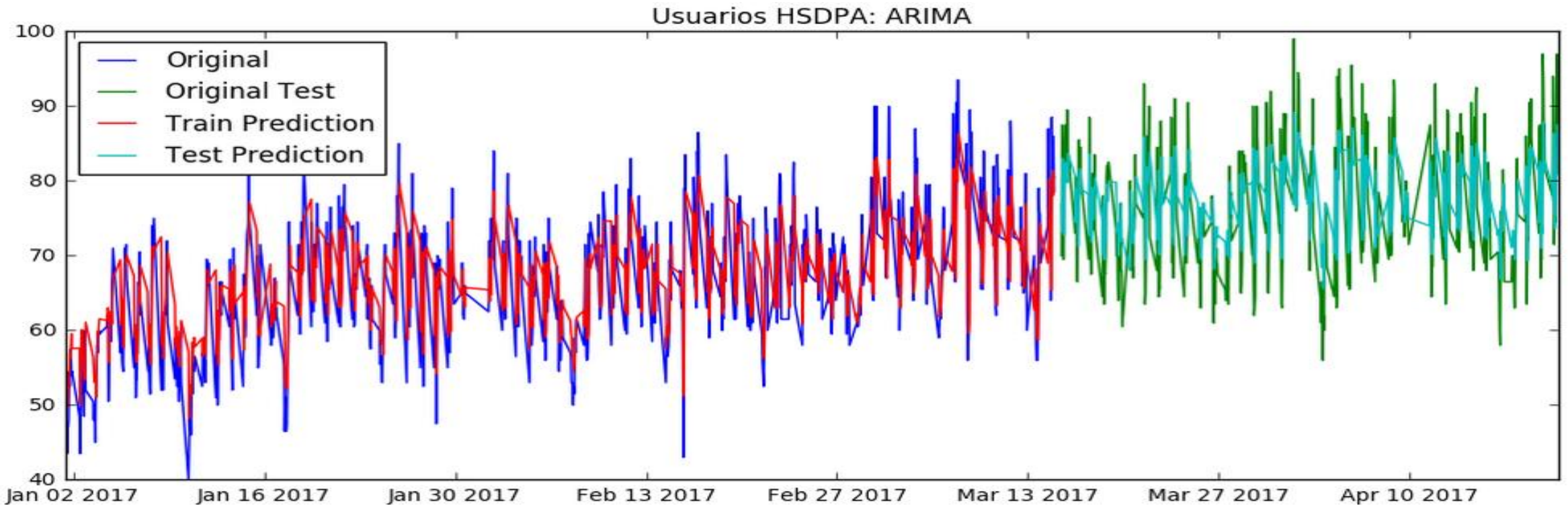
Akaike Information Critera (AIC) é uma medida de um modelo estatístico amplamente utilizada. O menor valor de AIC dentre os modelos indica o melhor modelo, qualitativamente, de acordo com:

1) o quão bom é o ajuste da curva
2) a simplicidade / parcimônia (menos é melhor)

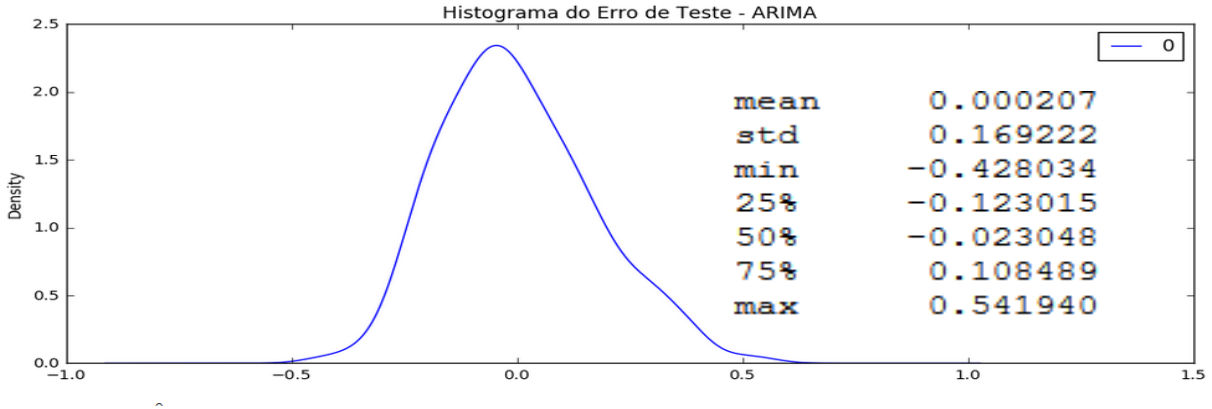
Considerando as funções de autocorrelação e correlação parcial anteriores e o valor de menor AIC, o modelo ARIMA(1,0,1) foi escolhido.

ARIMA(p,q)	p em AR(P)					
q em AR(Q)		1	2	3	4	5
	1	-1211.727	-1254.398	-1253.182	-1223.842	-1259.822
	2	-1238.993	-1253.597	-	-1259.717	-1259.290
	3	-1244.571	-1256.229	-1256.818	-1264.865	-1272.758
	4	-1254.216	-1257.647	-1266.030	-1289.770	-1253.271
	5	-1259.650	-1310.505	-1266.839	-1288.593	-1288.886

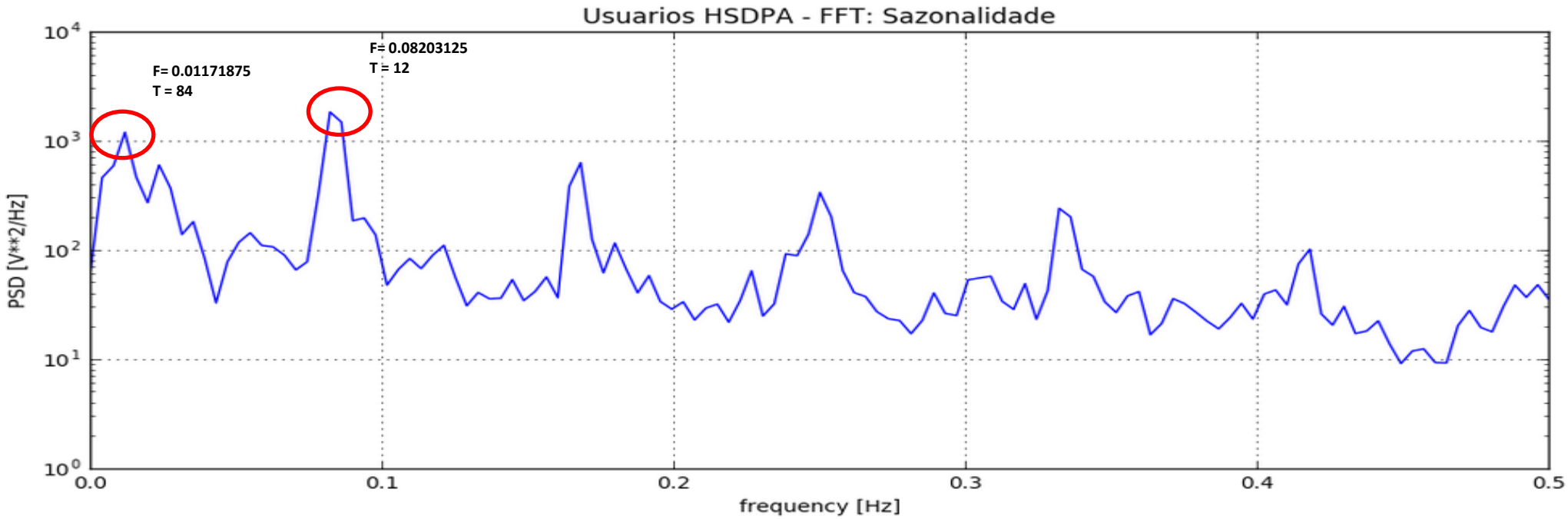
ARIMA – Previsão da séries



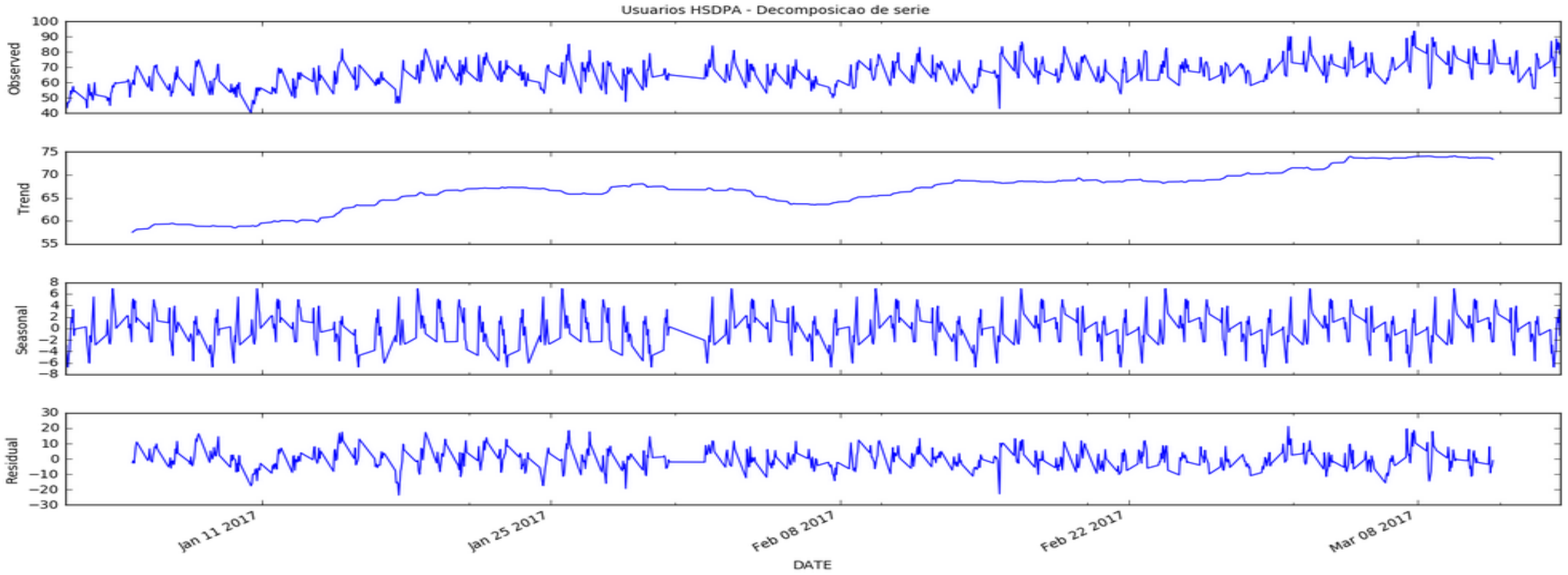
Fase	Algoritmo	MSE	RMSE	MAE	MAPE[%]
Treinamento	ARIMA(1,0,1)	**40.26**	**6.34**	**4.19**	**7.53**
Teste	ARIMA(1,0,1)	**52.82**	**7.26**	**5.82**	**7.61**



PMCs – FFT: Análise de sazonalidade



PMCs – Decomposição da ST

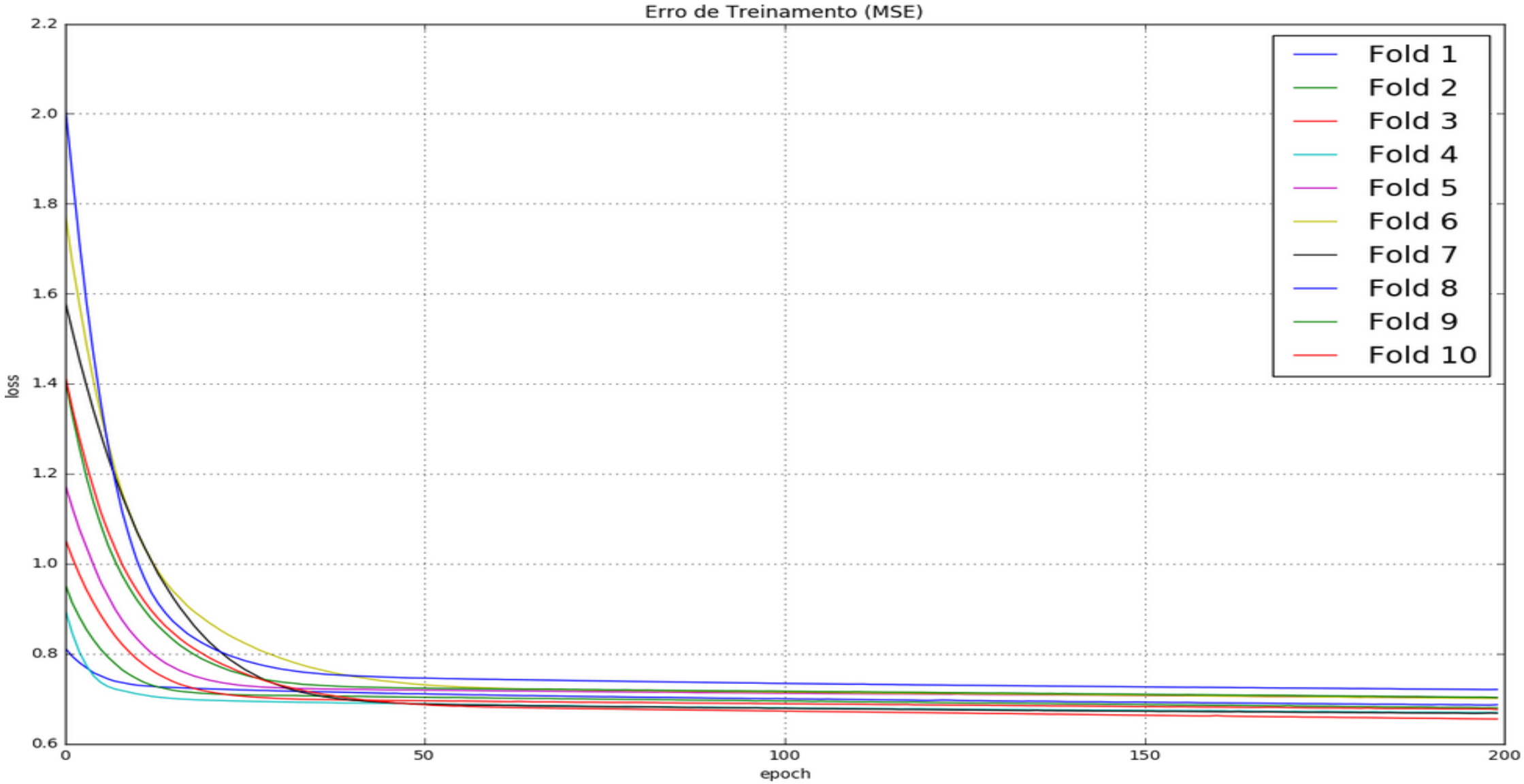


PMCs – Parâmetros e topogias

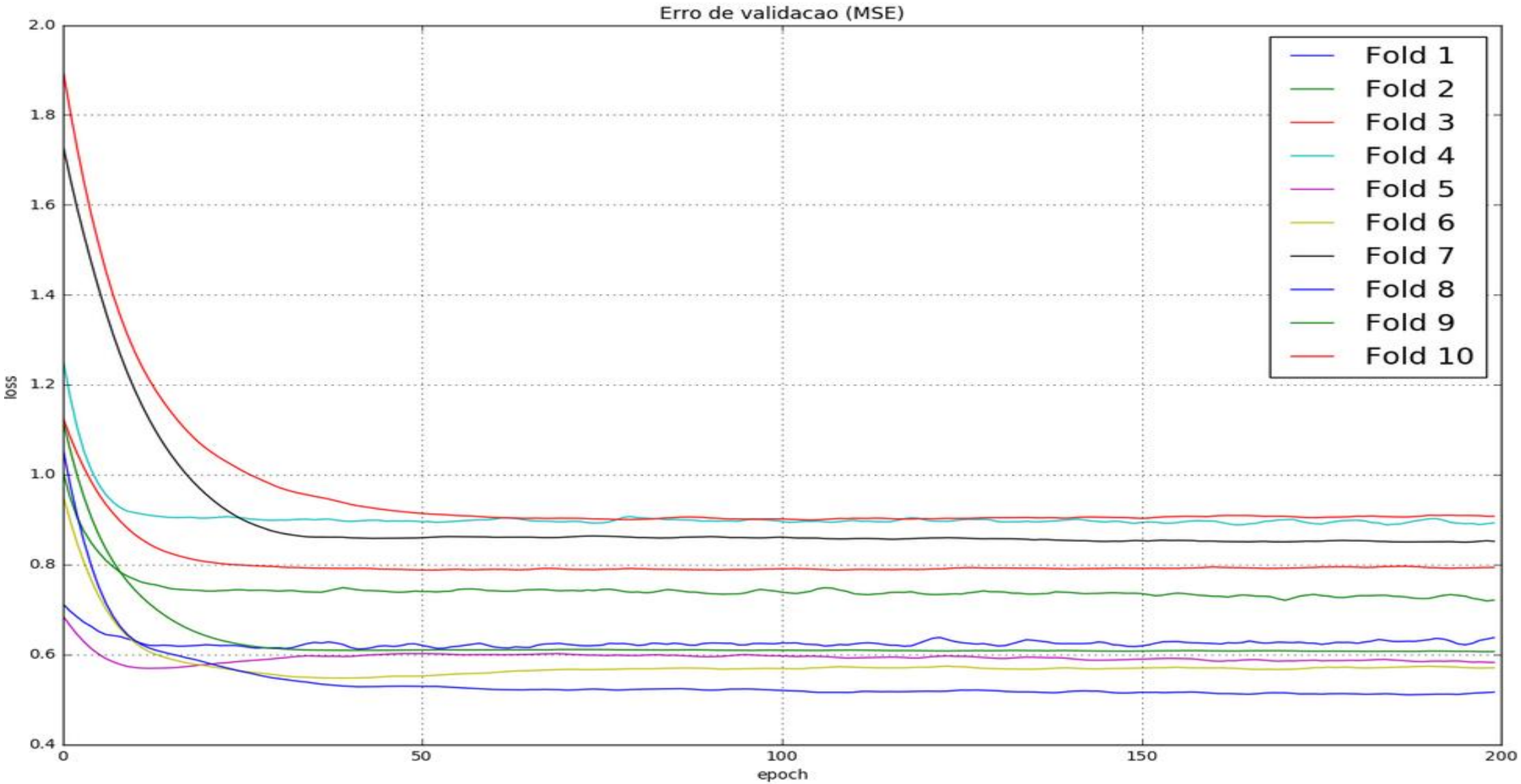
Parâmetros	Valores
Treinamento	Batch
Normalização	Média 0,desvio padrão 1
Epochs	15,200,400
Camada de Entrada	[1,2,3,4]
Camada de Intermediária	[4,8,12,24]
Camada de saída	1
Ativação Camada intermediária	Tanh
Ativação Camada de saída	Linear
Taxa de aprendizado	0.01
Momento	Não
Otimizador	SGD,Adam
Função Objetivo	MSE

Fase	Entrada	Camada Escondida	MSE	MAE	t(s)
Train	4	24	0.66443064	0.633254529	49.26199985
Train	4	8	0.66959015	0.633496375	52.16599989
Train	4	12	0.668339526	0.634134868	49.31599998
Train	4	24	0.689114527	0.641279221	13.78499985
Train	4	4	0.684734273	0.643728727	48.8039999
Train	3	4	0.69310642	0.645602808	50.12599993
Train	3	8	0.693184518	0.646302925	55.98100019
Train	3	24	0.702653597	0.646564	13.61300015
Train	3	24	0.693573354	0.646725996	51.95200014
Train	3	12	0.695044533	0.647048557	52.6960001
Train	4	8	0.693317068	0.647388133	13.35299993
Train	3	8	0.70556116	0.649542769	13.96500015
Train	1	5	0.71641991	0.650716314	194.6859999
Train	1	10	0.715563813	0.651798362	202.7120001
Train	2	12	0.704371876	0.65181259	55.28499985
Train	1	11	0.716102922	0.652049625	202.503
Train	1	8	0.715838921	0.652101496	57.31100011
Train	1	12	0.71580209	0.652152277	198.494
Train	1	7	0.715776916	0.65264086	199.9980001
Train	1	8	0.715578771	0.652828903	197.3710001
Train	2	24	0.710322411	0.652888051	13.59599996
Train	1	4	0.718290273	0.652956188	197.7509999
Train	2	24	0.705747139	0.652971368	52.65100002
Train	2	12	0.707794618	0.653268885	13.01499987
Train	2	4	0.704041132	0.653401503	53.5769999
Train	1	9	0.716143363	0.653456879	199.118
Train	2	8	0.708533925	0.653514268	53.26800013
Train	1	6	0.716273368	0.653975421	201.8380001
Train	1	8	0.723091909	0.654676064	12.87199998
Train	1	4	0.718556904	0.654676774	52.57300019
Train	3	12	0.71127453	0.654719657	16.73599982
Train	1	24	0.72788125	0.654836471	12.4230001
Train	2	8	0.712739858	0.654904667	13.74699998
Train	1	24	0.717632553	0.654949523	60.60599995
Train	1	3	0.717719095	0.654962245	198.5680001

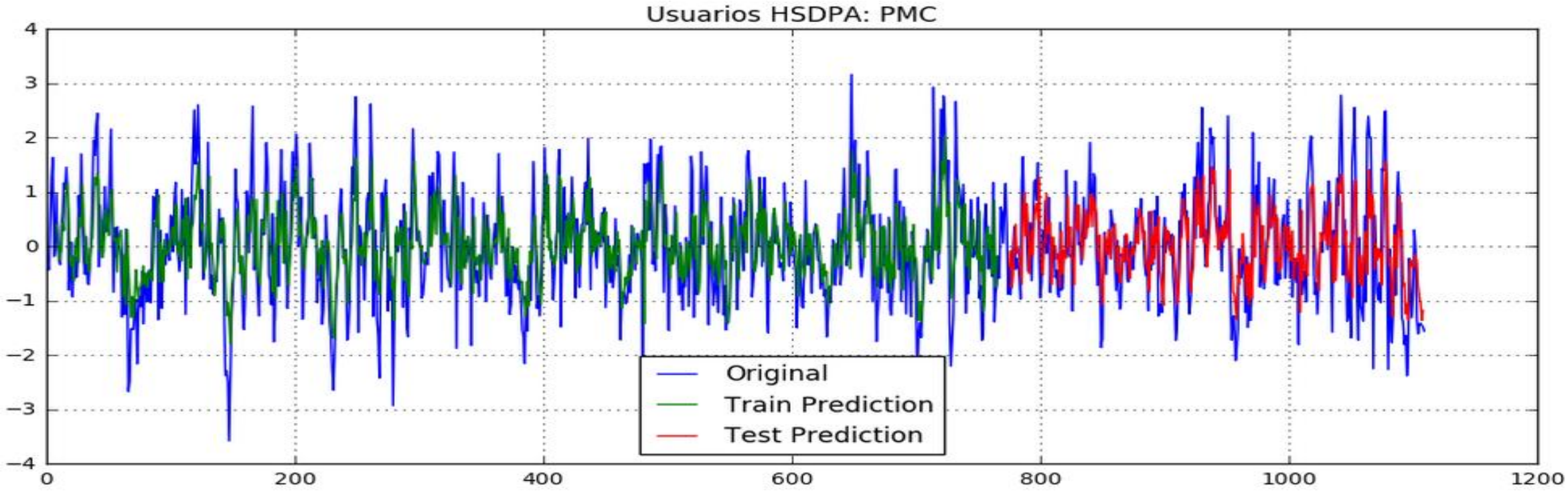
PMCs – Validação cruzada 10fold



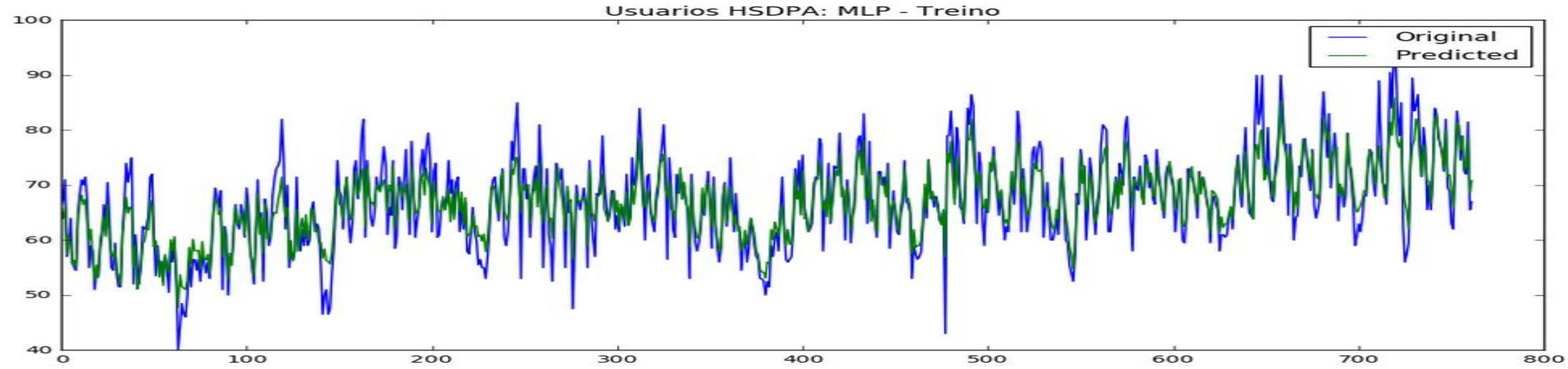
PMCs - Validação cruzada 10fold

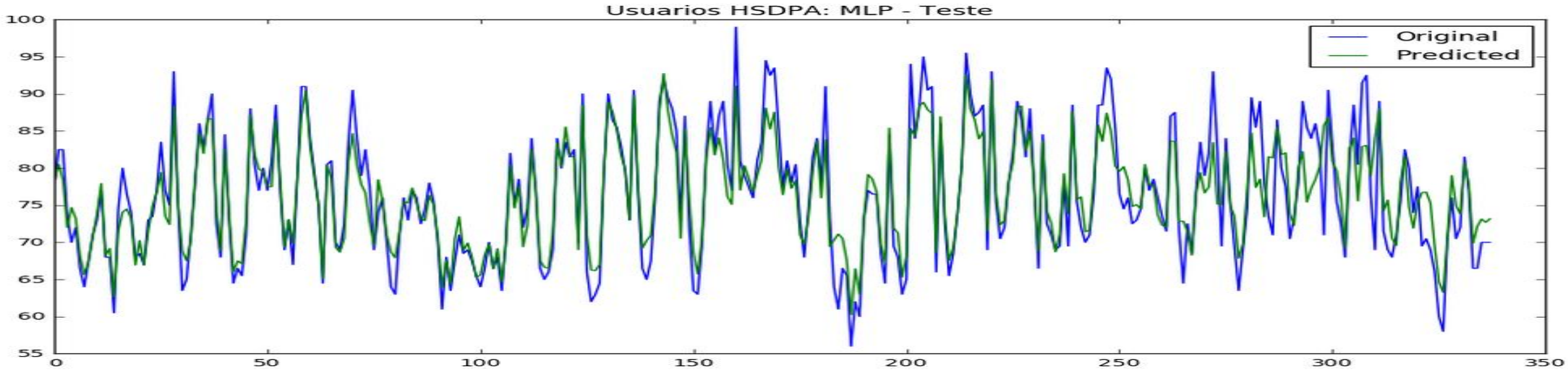


PMCs – Previsão da série

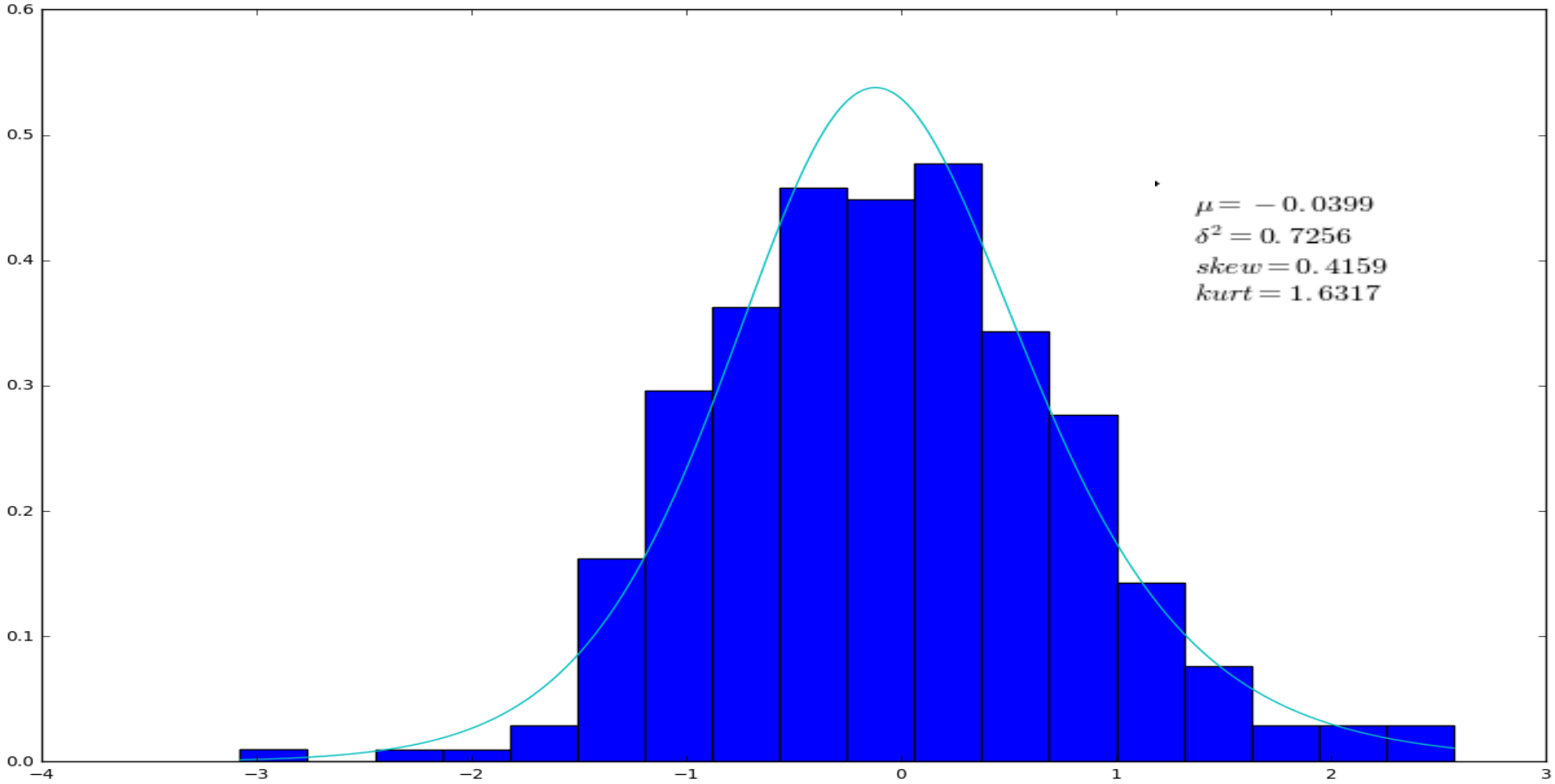


PMCs – Previsão da série





PMCs – Histograma do erro



PMCs – Corredores de validação

Para gerar os corredores validação, partiu-se do princípio de que os outliers podem ser identificados através da análise do erro de previsão **e(T) = x(T) - x~(T),** onde x ~(T) é o valor predito pelo
modelo para o instante T e x(T) é o valor coletado pelo sistema nesse instante.

Intuitivamente, sabe-se que, se este **erro é grande o suficiente, podemos concluir que a observação x(T) contém erro ou foi gerada por um processo diferente**. Dessa forma, o teste de verificação de outliers assume forma:

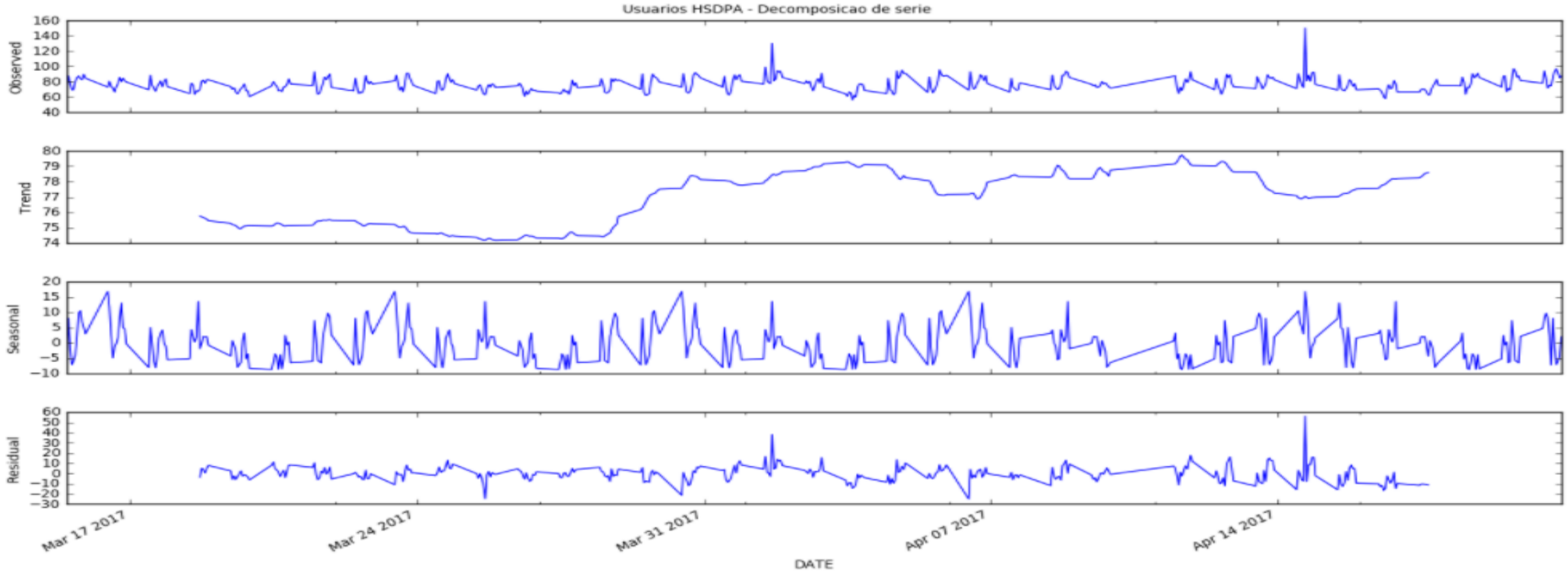
$$\left|\frac{e(T)}{\hat{\sigma}_e}\right| > k$$

Onde tipicamente 4 < k < 5, tendo seu valor ajustado de acordo com a série e com o modelo desenvolvido. O valor sigma é correspondende ao valor RMS da distribuição dos valores do erro e(T) acumulados até o instante (t-1). [4] Assim, a largura total do corredor vale 2k*sigmae, e os limites são:

xI(T) = x~(T) – k*sigmae
xS(T) =x~(T) + k*sigmae

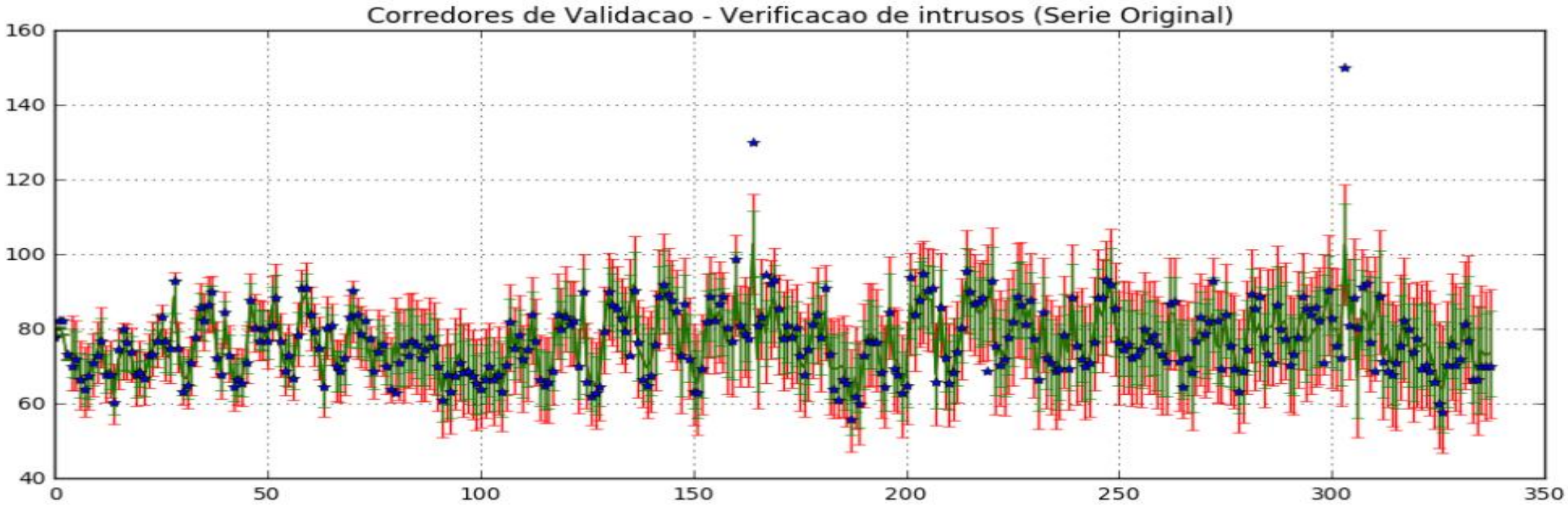
PMCs – Corredores de validação

2 Outliers foram adicionados à série para verificação dos corredores.



PMCs – Corredores de validação

Corredores de validação utilizando k=2 e k=3.
Os pontos da série abaixo correspondem ao valor real medido, presente no dataset de teste.



Avaliação

MSE - Mean Squared Error
RMSE - Root Mean Squared Error
MAE - Mean Absolute Error
MAPE - Mean Absolute Percentage Error

Fase	Algoritmo	MSE	RMSE	MAE	MAPE[%]
Treinamento	ARIMA(1,0,1)	**40.26**	**6.34**	**4.19**	**7.53**
Teste	ARIMA(1,0,1)	**52.82**	**7.26**	**5.82**	**7.61**
Treinamento	MLP(4,24,1)	**11.96**	**3.45**	**2.6**	**3.95**
Teste	MLP(4,24,1)	**12.19**	**3.49**	**2.55**	**3.35**

Conclusão

• Este trabalho apresentou um estudo sobre modelagem e previsão de séries temporais no contexto de modelagem de séries de indicadores (KPIs) de um rede de telefonia móvel.

• Foi proposto uma metodologia de qualidade de dados para o contexto de bases de dados de indicadores de rede móvel.

• A rede PMCs apresentou desempenho superior à modelagem clássica ARIMA.

• Para criação de corredores de validação foi utilizado o valor RMS da distribuição dos valores do erro e(T) acumulados até o instante (t-1).

• Em trabalho futuros
    • datasets maiores devem ser utilizados;
    • prever valores futuros de longo prazo;
    • testar diferentes arquiteturas de redes, como redes recorrentes.
    • Utilizar séries adicionais que apresentem correlação cruzada na modelagem das séries alvo.
    • Testar diferentes metodologias de corredores de validação em séries que contenham intrusos conhecidos.

Referências

[1] Calôba L. P. – Introdução ao Uso de Redes Neurais na Modelagem de Sistemas Dinâmicos e Séries Temporais, Livro de Minicursos do XIV Congresso Brasileiro de Automática, Natal, 2002.

[2] Uso de um processo ETL em um modelo Data Warehouse para a geração de dashboards de indicadores de redes de telefonia celular

[3] 3GPP ”Specification”. 2015. Disponível em: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2026

[3] Dantas A. C. H. – Sistema de Monitoração de Qualidade de Dados, Tese de Doutorado.

[4] Almeida,D.M., Daniel C. Cunha,D.C., Neto,P. S. G. M., A Proposal of an Intelligent Forecasting System for Automotive Diagnostics using Time Series Analysis. UFPE.

[5] Guedes,E. B., Lima, P. M., Oliveira, M. B. L.. Neural Networks for Time Series Rainfall Forecasting: A Case Study in Manaus, Amazonas. Universidade Estadual do Amazonas.

[6] Dong, X.,Fan, W., Gu,J. Predicting LTE Throughput Using Traffic Time Series. Beijing University of Posts and Telecommunications

[7] Adhikari, R.,Agrawal, R.K., An Introductory Study on Time Series Modeling and Forecasting.

Obrigado

Carlos Eduardo Covas Costa
kaducovas@gmail.com
https://github.com/kaducovas/timeSeries