

Resultados com o método descrito por Zhang:

Conteúdo

1	Resultados do artigo	2
2	Proposta 1	3
2.1	Dígitos Significativos	3
2.2	Algumas observações	15
2.3	Influência de k	17
3	Proposta 2	18
3.1	Dígitos Significativos	18
3.2	Algumas observações	34

1 Resultados do artigo

Série temporal usada é o mapa logístico com $R = 4$ e $x_0 = 0.36$.

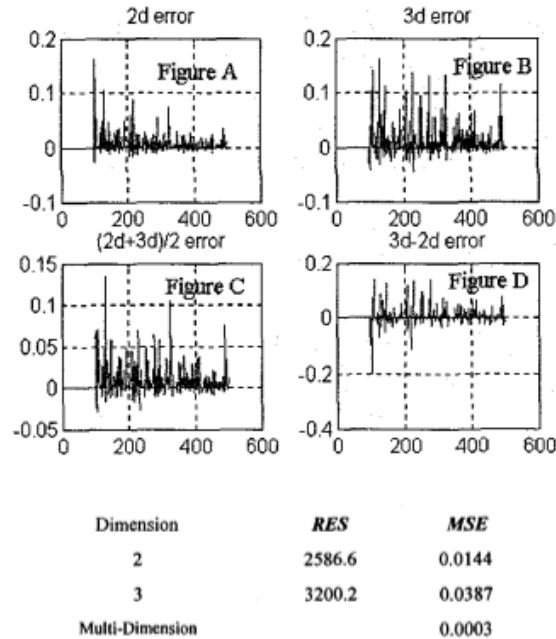


Figura 1: Resultados obtidos por Zhang

Pelos gráficos, temos a impressão que são dados os 100 primeiros pontos e o objetivo é prever os próximos 412 pontos da série. No entanto, ainda não fica claro se os pontos previstos influenciam nos próximos a serem previstos. Dessa forma, duas propostas de algoritmo são feitas:

Proposta 1: Se o ponto x_n está sendo previsto pelo método, todos os anteriores são levados em consideração para eleger o ponto de mínima distância (isto é, inclui os que foram calculados pelo método).

Proposta 2: Os novos pontos são comparados apenas com os 100 da série "original", novos pontos obtidos com o método não servirão como base de comparação para achar o ponto de distância mínima.

Dentro dessas propostas também serão exploradas outras variáveis como precisão e influência de k .

2 Proposta 1

2.1 Dígitos Significativos

dígitos significativos	(2D) mae	(2D) mse	(2D) res
15	0.399654198946355	0.240932279587018	29569.9152005802
14	0.39561267842643	0.23751451365323	8004.180597727
13	0.4175085199098	0.2566241372111	117854.7829951
12	0.37229996298	0.219768275115	6993.64271591
11	0.35667599699	0.21018351354	430206.96943
10	0.3358954898	0.1841327421	61429.05733
9	0.460179874	0.299810366	26147.8885
8	0.41524571	0.25348344	8780.1178

Tabela 1: Erros usando embedding 2D

dígitos significativos	(3D) mae	(3D) mse	(3D) res
15	214720.965400429	456182813888.252	1396814330.72686
14	0.54371135215549	0.45216772482027	8495.2261716475
13	17678.39479885	3093132566.003	897638207.351
12	0.454491310867	0.327924824084	7063.56144771
11	76580.129101	58038598133.0	415412483.79
10	0.3878222847	0.2334424212	64498.7866
9	0.419867047	0.277871999	45962.5778
8	4052.3754	1.6262277e+8	9880324.3

Tabela 2: Erros usando embedding 3D

dígitos significativos	(MD) mae	(MD) mse	(MD) res
15	107360.529031089	114045711592.789	698421862.704213
14	0.39161721565719	0.23218254852685	8193.3856277685
13	8839.245060138	773283669.5028	448877934.4927
12	38290.126014	14509651524.0	207921274.87
11	0.373989499292	0.215523809099	6992.67047025
10	0.3382768714	0.1674416298	62945.12003
9	0.383053387	0.212413185	36013.8018
8	2026.2326	40655666.0	4944456.9

Tabela 3: Erros usando embedding multidimensional

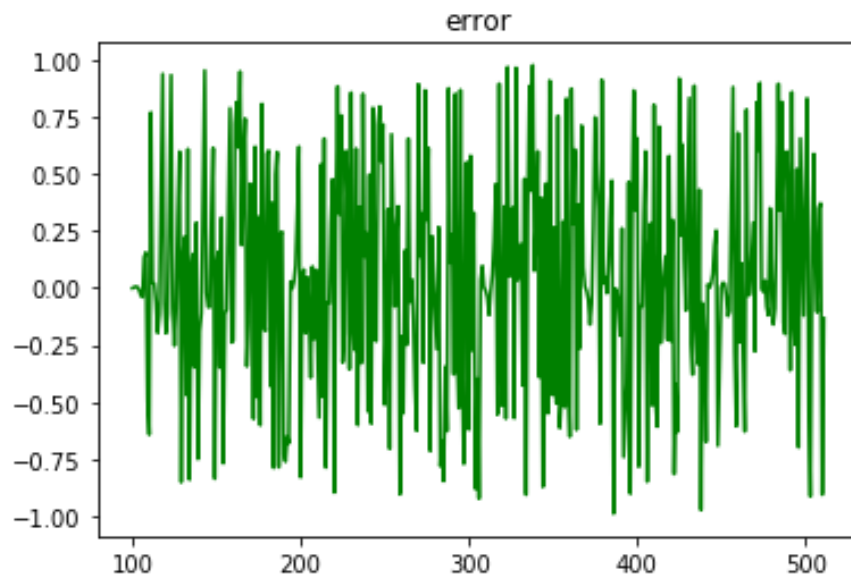


Figura 2: Erro com precisão de 15 dígitos usando embedding em 2D

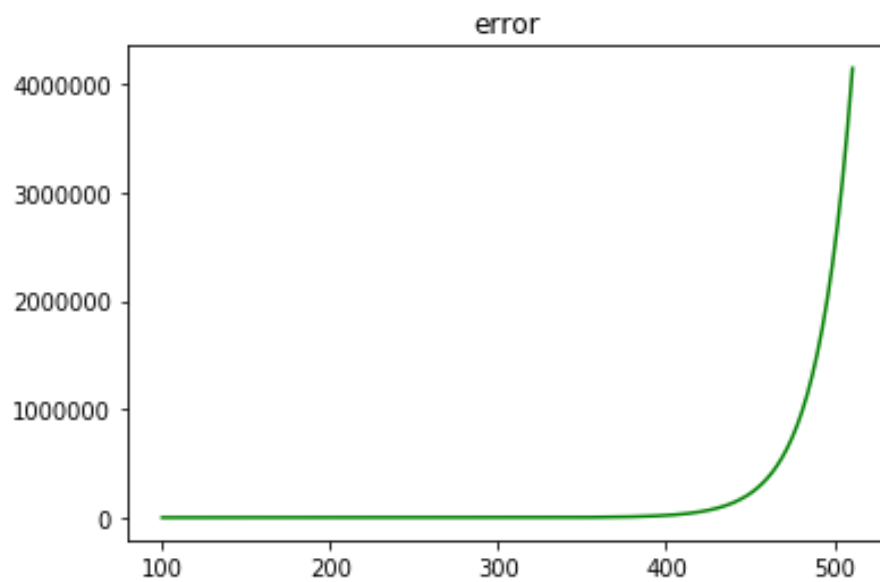


Figura 3: Erro com precisão de 15 dígitos usando embedding em 3D

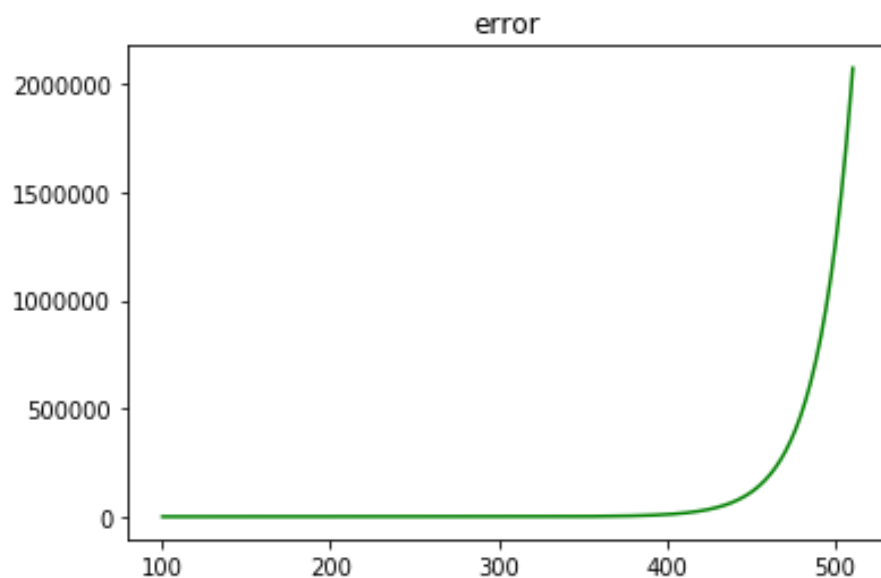


Figura 4: Erro com precisão de 15 dígitos usando embedding multidimensional

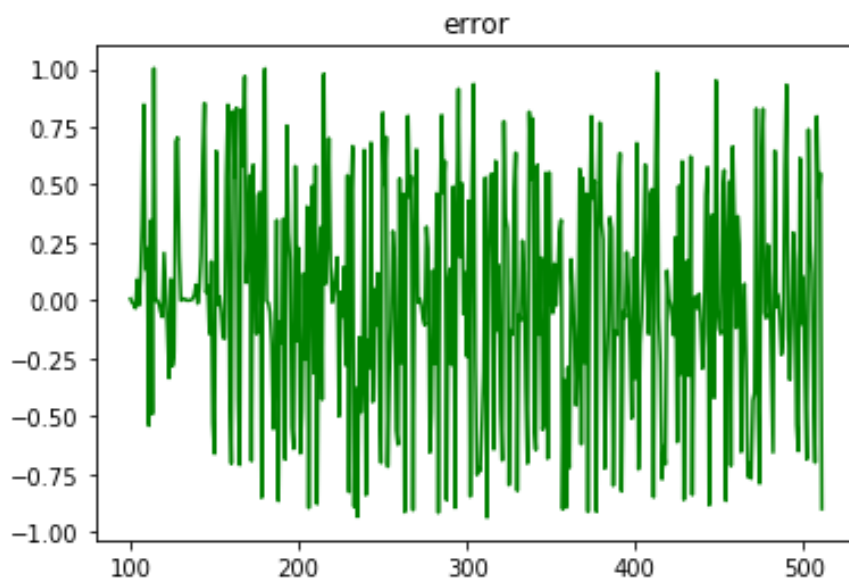


Figura 5: Erro com precisão de 14 dígitos usando embedding em 2D

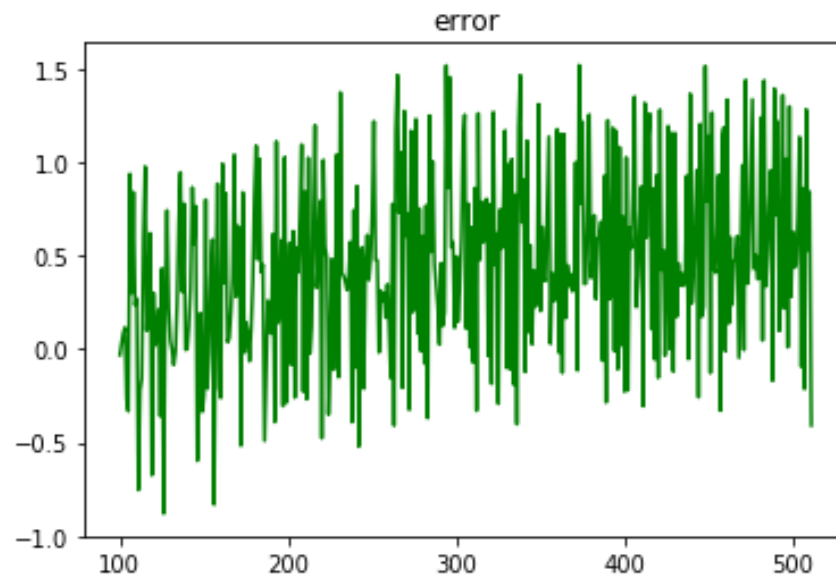


Figura 6: Erro com precisão de 14 dígitos usando embedding em 3D

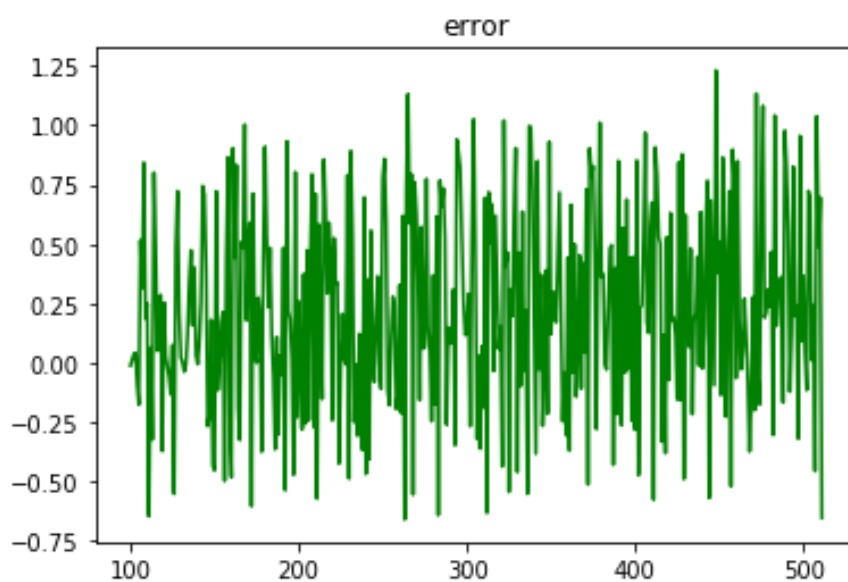


Figura 7: Erro com precisão de 14 dígitos usando embedding multidimensional

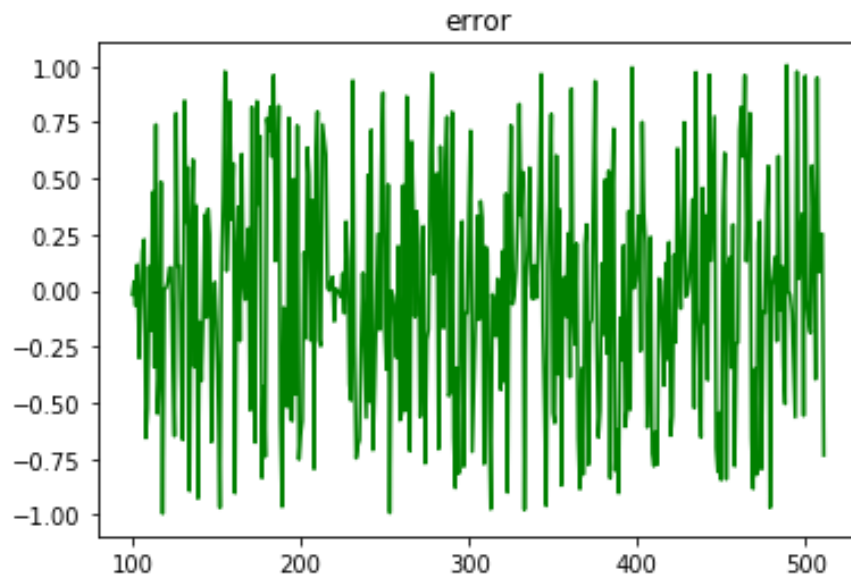


Figura 8: Erro com precisão de 13 dígitos usando embedding em 2D

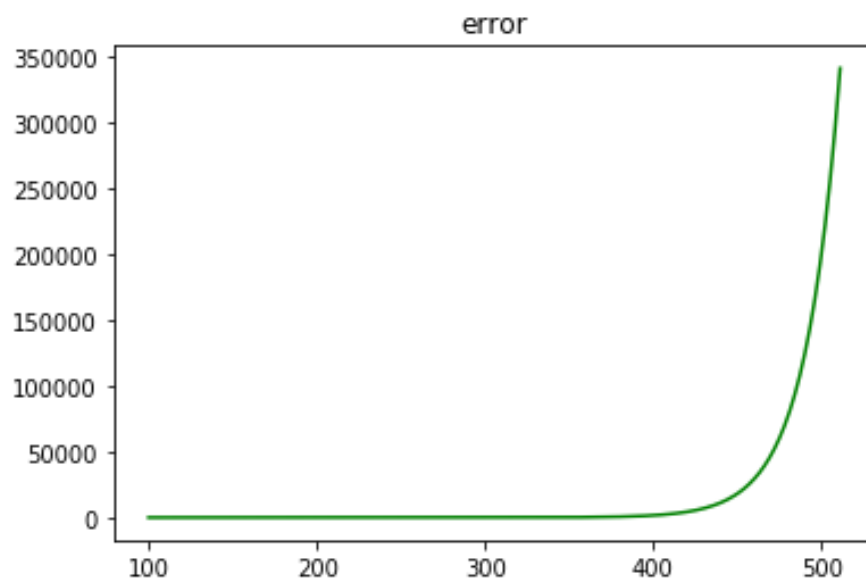


Figura 9: Erro com precisão de 13 dígitos usando embedding em 3D

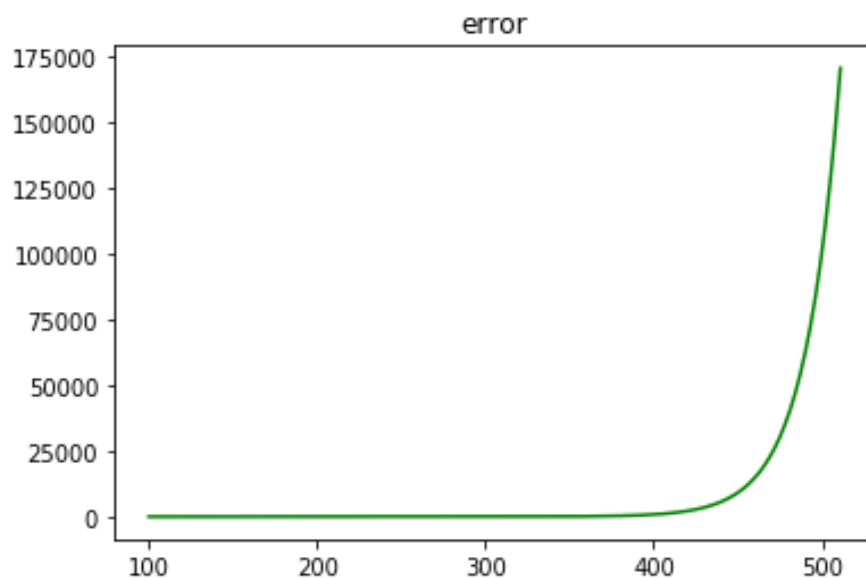


Figura 10: Erro com precisão de 13 dígitos usando embedding multidimensional

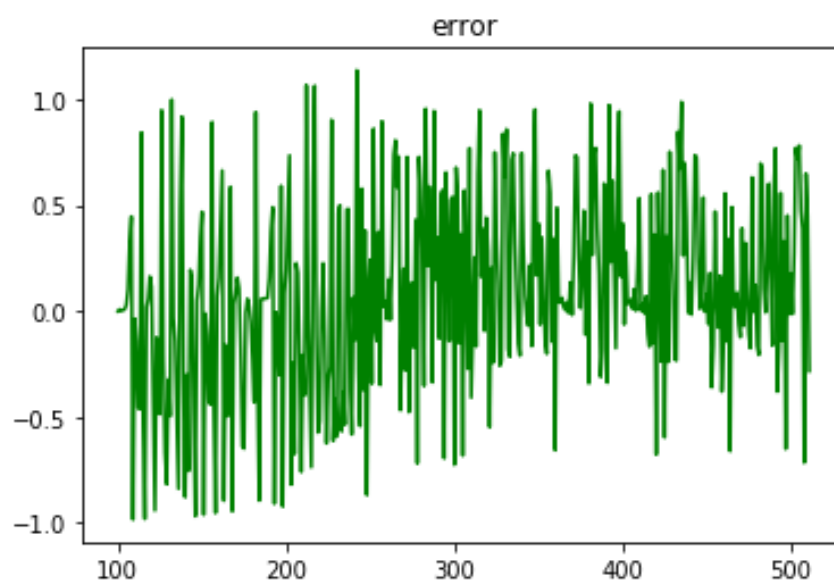


Figura 11: Erro com precisão de 12 dígitos usando embedding em 2D

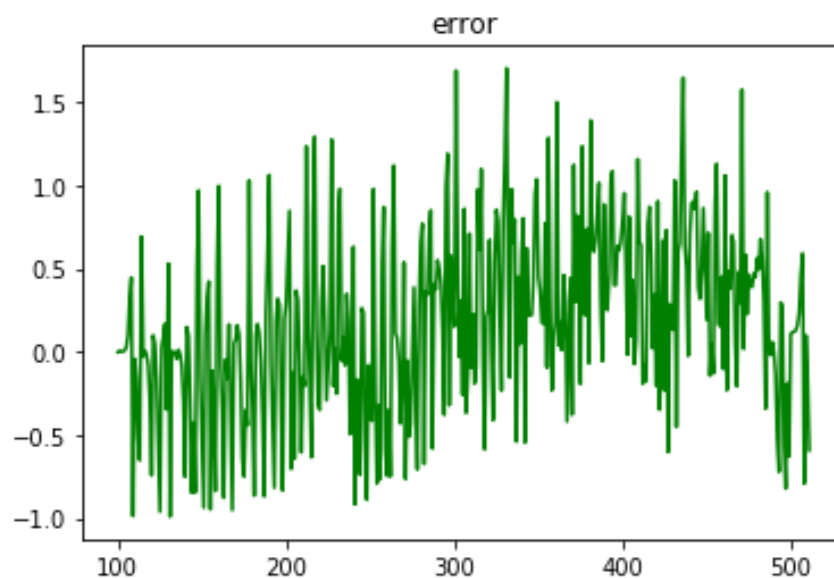


Figura 12: Erro com precisão de 12 dígitos usando embedding em 3D

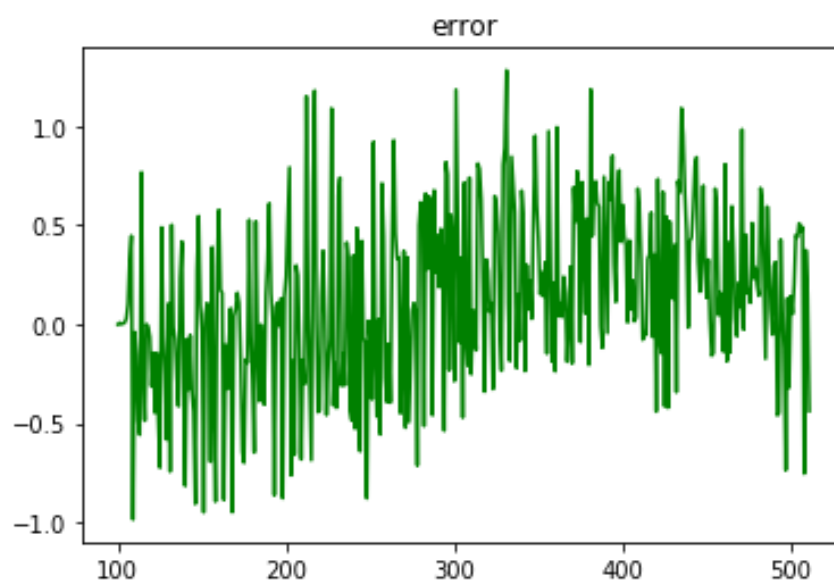


Figura 13: Erro com precisão de 12 dígitos usando embedding multidimensional

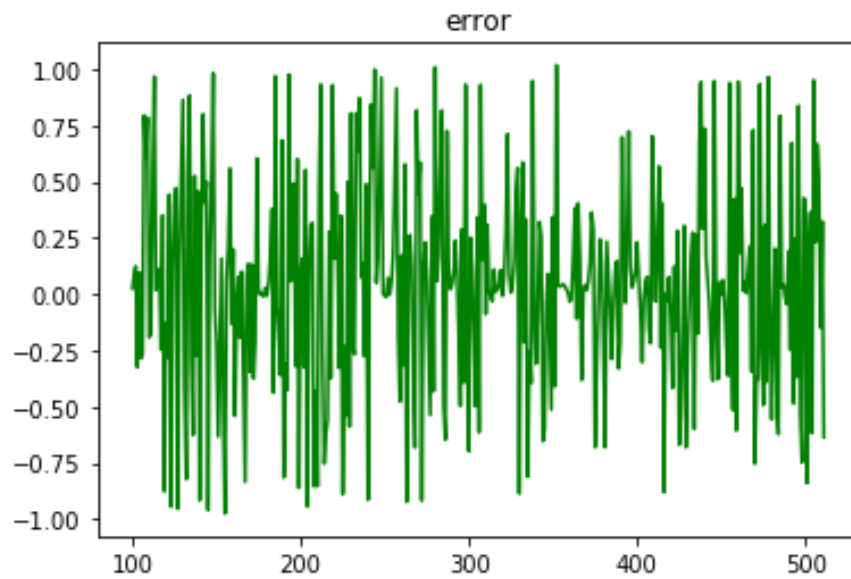


Figura 14: Erro com precisão de 11 dígitos usando embedding em 2D

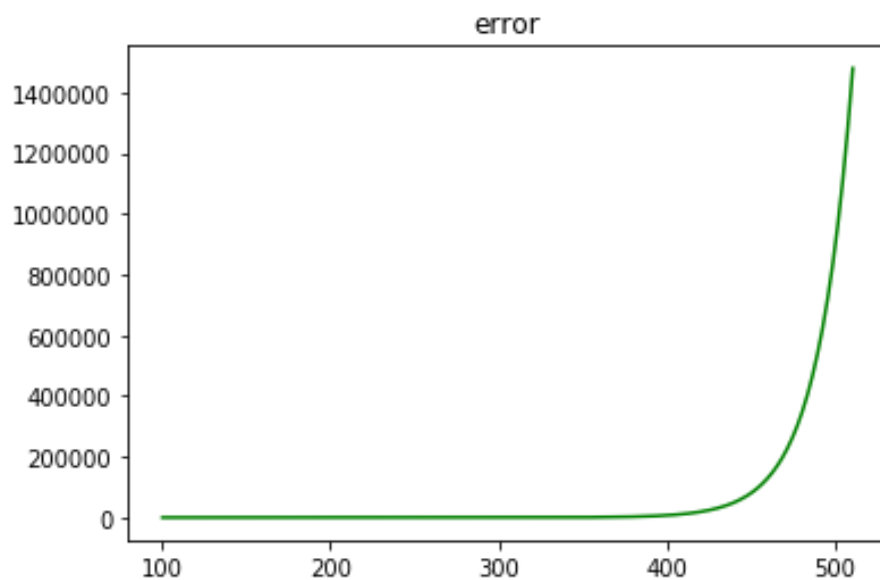


Figura 15: Erro com precisão de 11 dígitos usando embedding em 3D

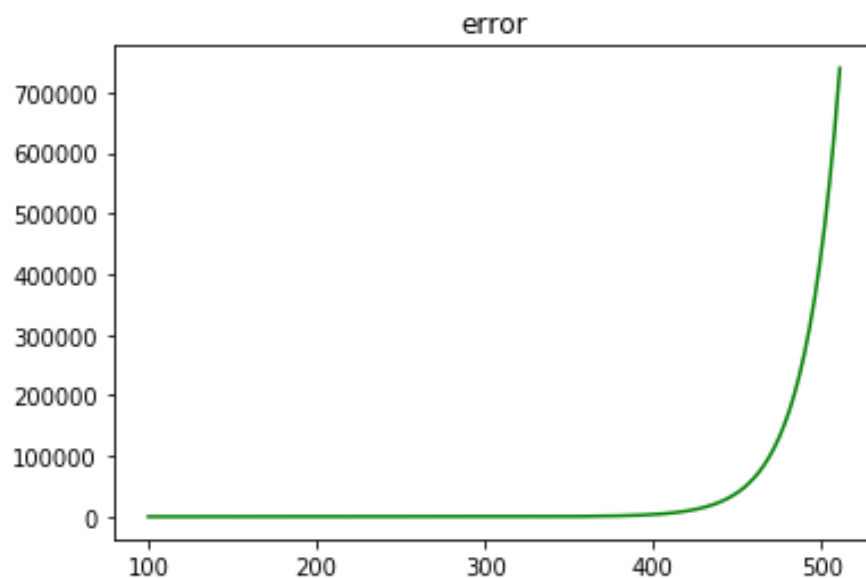


Figura 16: Erro com precisão de 11 dígitos usando embedding multidimensional

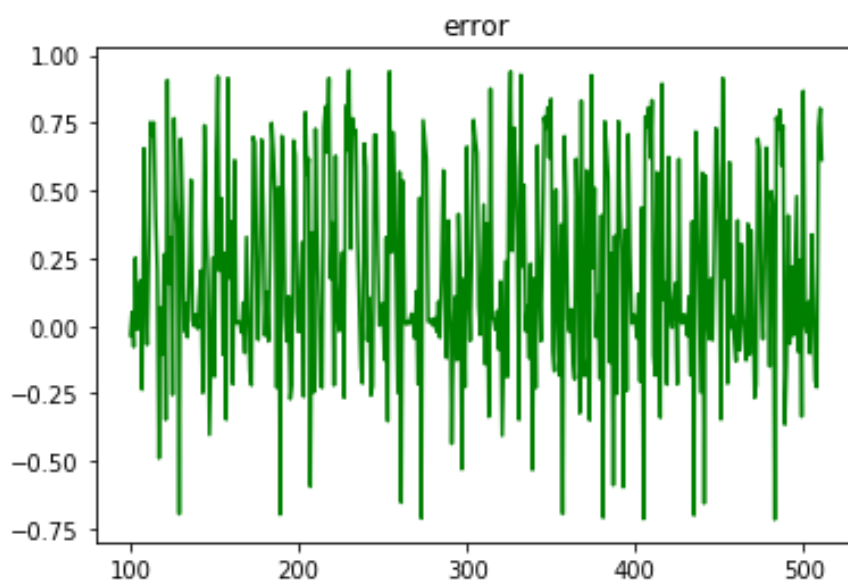


Figura 17: Erro com precisão de 10 dígitos usando embedding em 2D

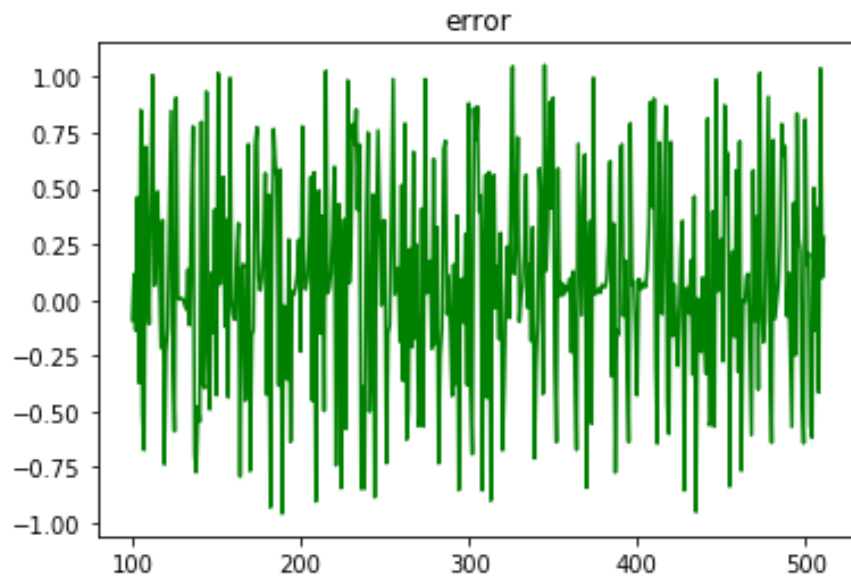


Figura 18: Erro com precisão de 10 dígitos usando embedding em 3D

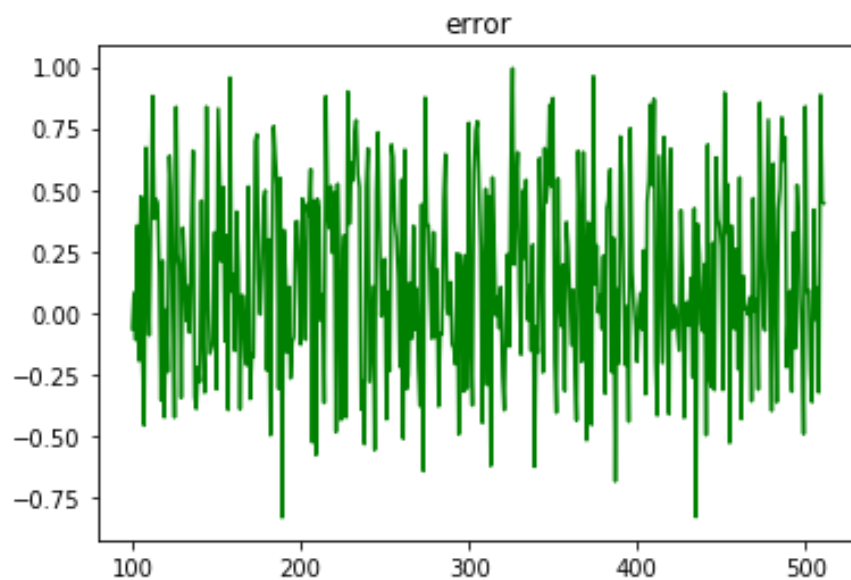


Figura 19: Erro com precisão de 10 dígitos usando embedding multidimensional

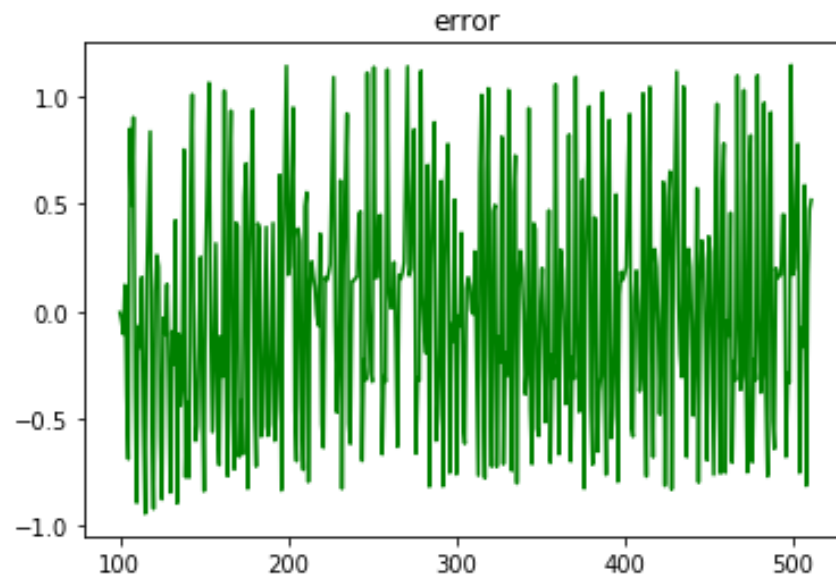


Figura 20: Erro com precisão de 9 dígitos usando embedding em 2D

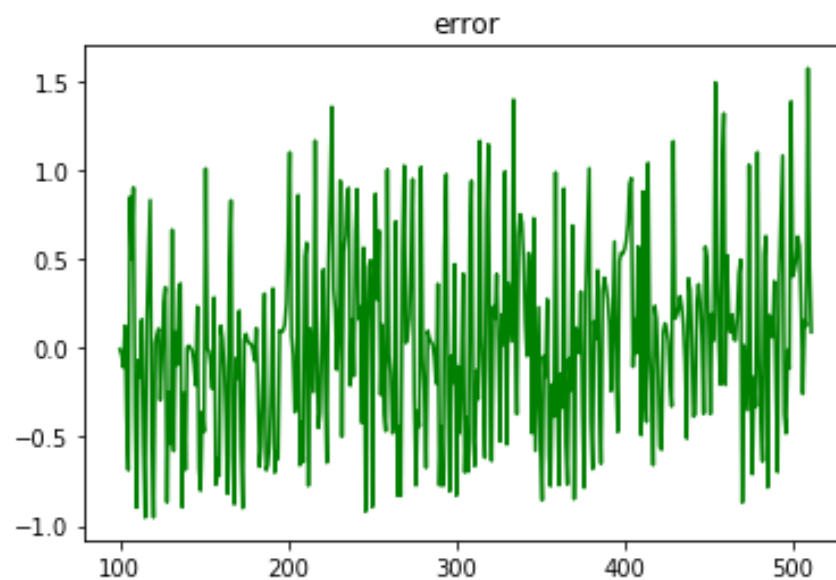


Figura 21: Erro com precisão de 9 dígitos usando embedding em 3D

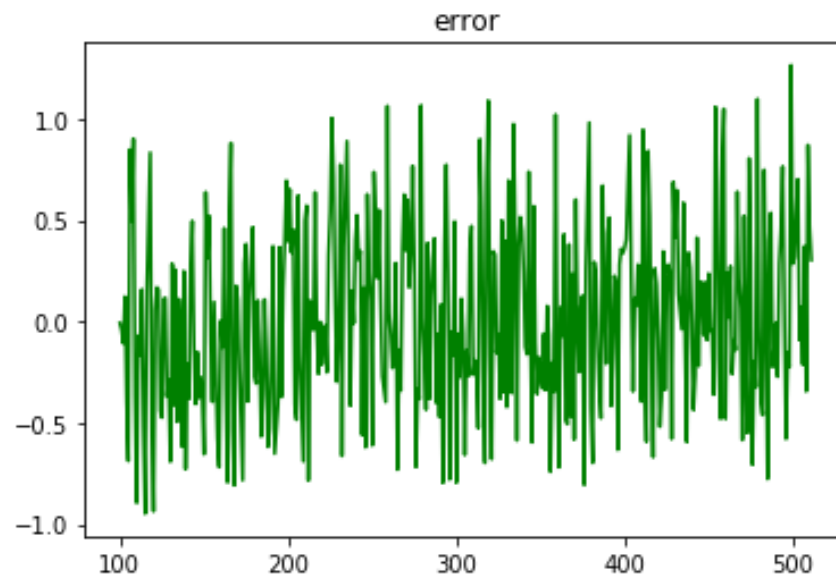


Figura 22: Erro com precisão de 9 dígitos usando embedding multidimensional

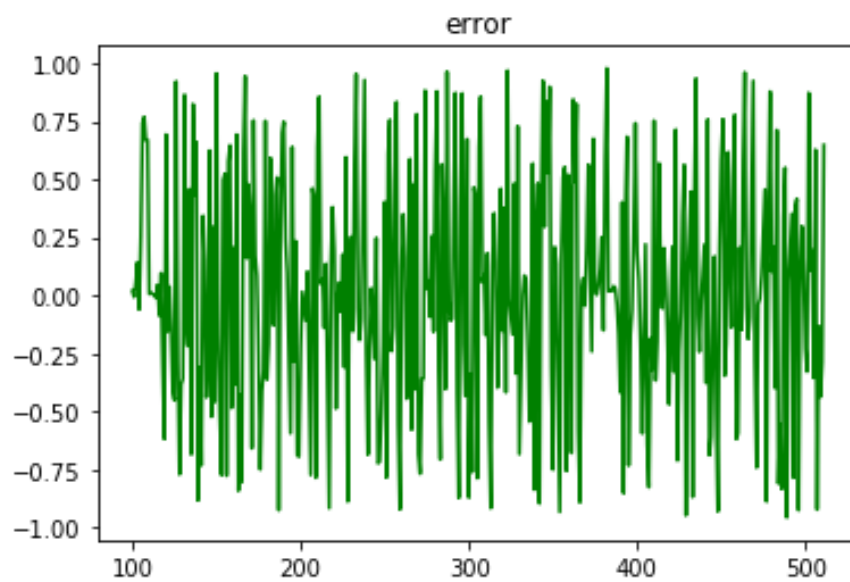


Figura 23: Erro com precisão de 8 dígitos usando embedding em 2D

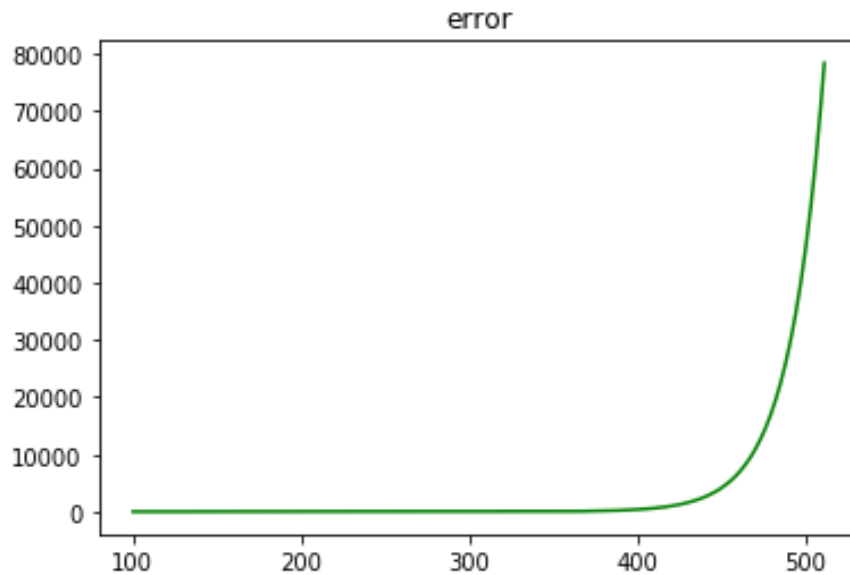


Figura 24: Erro com precisão de 8 dígitos usando embedding em 3D

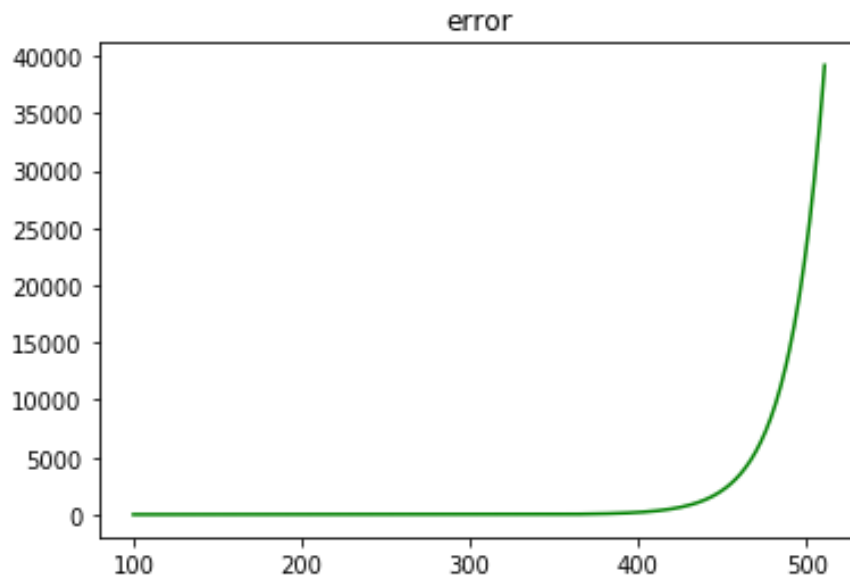


Figura 25: Erro com precisão de 8 dígitos usando embedding multidimensional

2.2 Algumas observações

Nota-se um fenômeno interessante: para algumas precisões, a previsão da série temporal para o embedding 3d (e multidimensional, por consequência) cresce de forma exponencial. Nesse contexto, duas perguntas naturalmente são feitas:

- 1) em que ponto o erro foge do intervalo $[-1,1]$?
- 2) se continuássemos prevendo pontos com embedding 2D, uma forma exponencial ficaria evidente?

As próximas figuras nos auxiliam nessas questões.

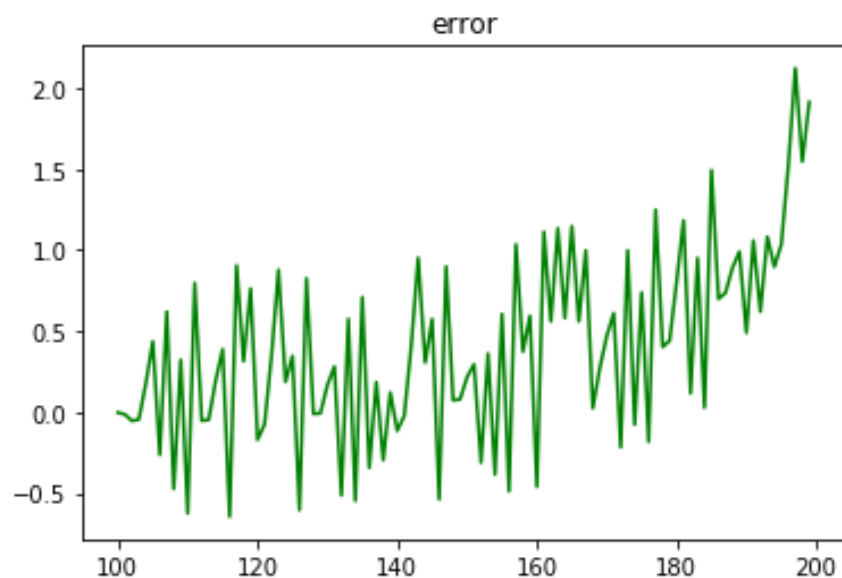


Figura 26: Erro com precisão de 15 dígitos usando embedding 3D

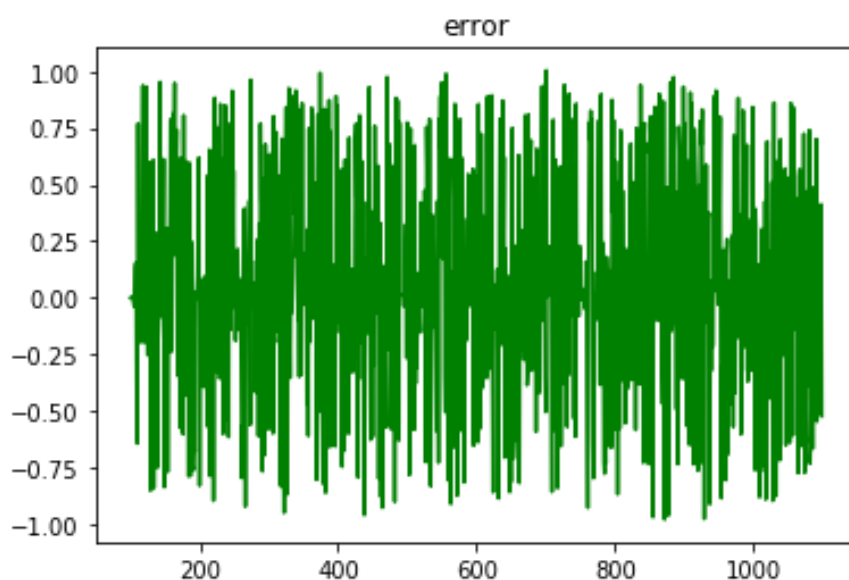


Figura 27: Erro com precisão de 15 dígitos usando embedding 2D

Com o embedding 3D, a série foge do intervalo esperado em menos de 100 iterações, enquanto com 2D, a série não dá qualquer sinal de comportamento exponencial, mesmo aumentando as iterações até 1000. Tal fato indica que essa divergência não tem relação com o quão longe se tenta prever o futuro de uma série temporal.

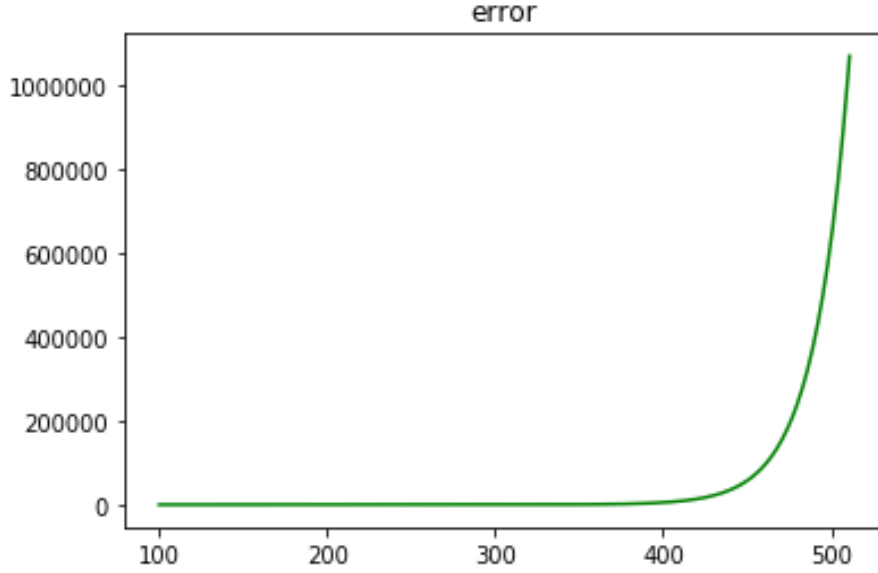


Figura 28: Erro com precisão de 15 dígitos usando embedding 2D

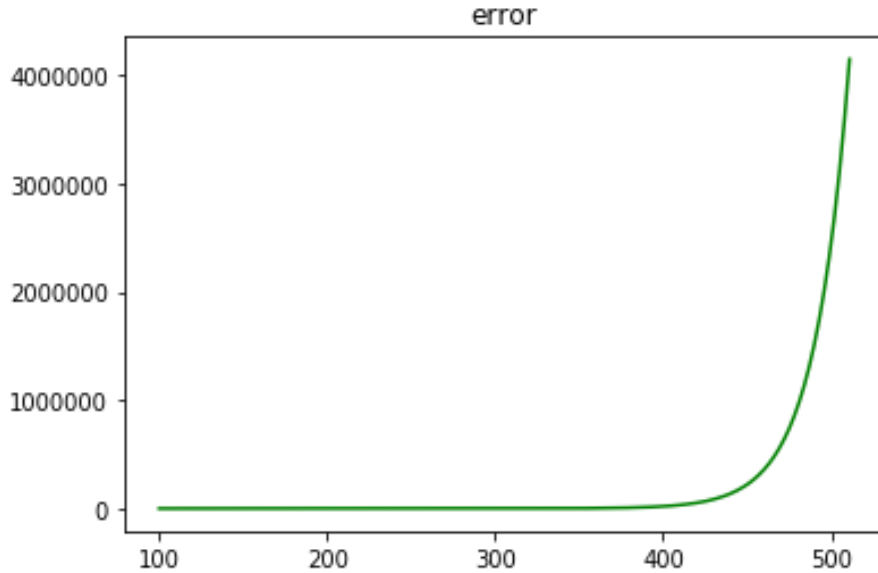


Figura 29: Erro com precisão de 15 dígitos usando embedding 3D

Não é difícil entender porque esse método é falho, uma vez que a função de dist1 (em relação a k) é exponencial. O erro nesses casos está na mesma ordem de grandeza do obtido no caso da proposta 1 com $k=1$ para alguns embeddings e precisões.

2.3 Influência de k

Até agora, estamos escolhendo sempre o ponto de mínima de distância do último ponto conhecido (x_{n-1}) e com base nisso, calculando dist1 , como descrito no artigo, sendo que k é sempre igual a 1. Agora tentaremos achar um ponto de distância mínima apenas no primeiro número da série e a partir dali, todos os outros serão calculados, aumentando-se o passo k .

3 Proposta 2

3.1 Dígitos Significativos

dígitos significativos	(2D) mae	(2D) mse	(2D) res
15	0.393401522446316	0.23904041973728	31446.7739244839
14	0.35805270249551	0.20662922002618	5499.1935757671
13	0.3751851017606	0.2218021997265	105776.329639
12	0.426968103737	0.287776576451	2814.86613265
11	0.40277062157	0.2529806459	802545.18094
10	0.3334419189	0.1812873911	60993.32579
9	0.443305425	0.278940928	64654.7535
8	0.39405846	0.23745588	2814.0381

Tabela 4: Erros usando embedding 2D

dígitos significativos	(3D) mae	(3D) mse	(3D) res
15	0.403254317803292	0.25415417569421	7184.47921495079
14	0.39390592267527	0.23972418779192	9529.9086504399
13	0.3831959449064	0.2242752138807	120617.5745435
12	0.361114171923	0.208220977871	6738.11895098
11	0.38718253422	0.23889127855	613775.68263
10	0.4134859489	0.2596410946	30389.12409
9	0.407583437	0.262124656	60658.5287
8	0.34195476	0.19962881	13535.548

Tabela 5: Erros usando embedding 3D

dígitos significativos	(MD) mae	(MD) mse	(MD) res
15	0.354196771793409	0.182795945901353	19264.7093844975
14	0.33648329126398	0.16707942173685	7478.9818145504
13	0.3430464882182	0.1732041958474	113162.3392344
12	0.343507574452	0.172187491403	4702.06151064
11	0.3416512854	0.18060659435	708102.56352
10	0.3416840332	0.1679145971	45662.74143
9	0.375490602	0.203789231	62597.8555
8	0.31906616	0.14033786	8123.3234

Tabela 6: Erros usando embedding multidimensional

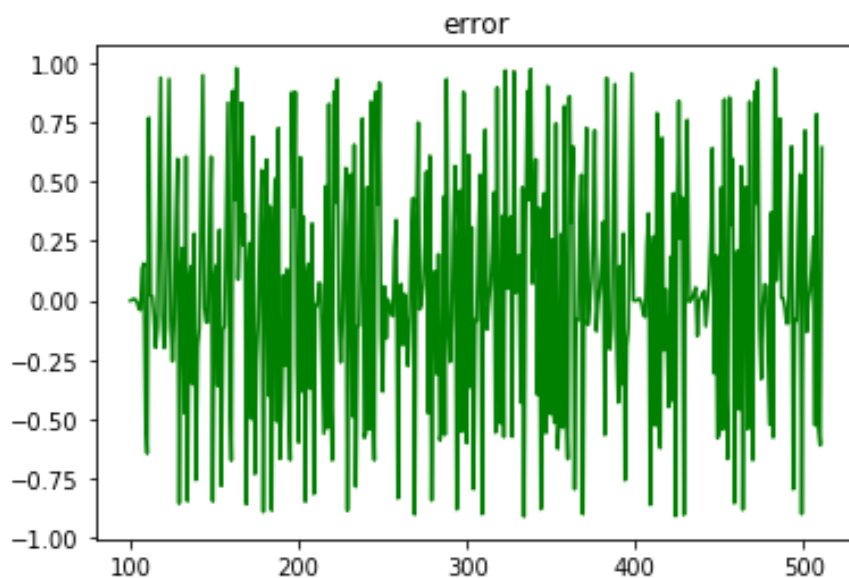


Figura 30: Erro com precisão de 15 dígitos usando embedding em 2D

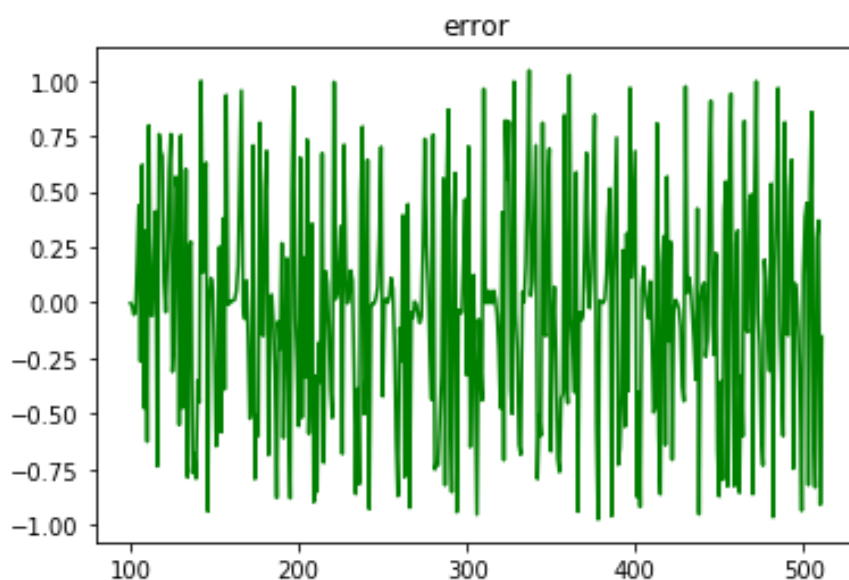


Figura 31: Erro com precisão de 15 dígitos usando embedding em 3D

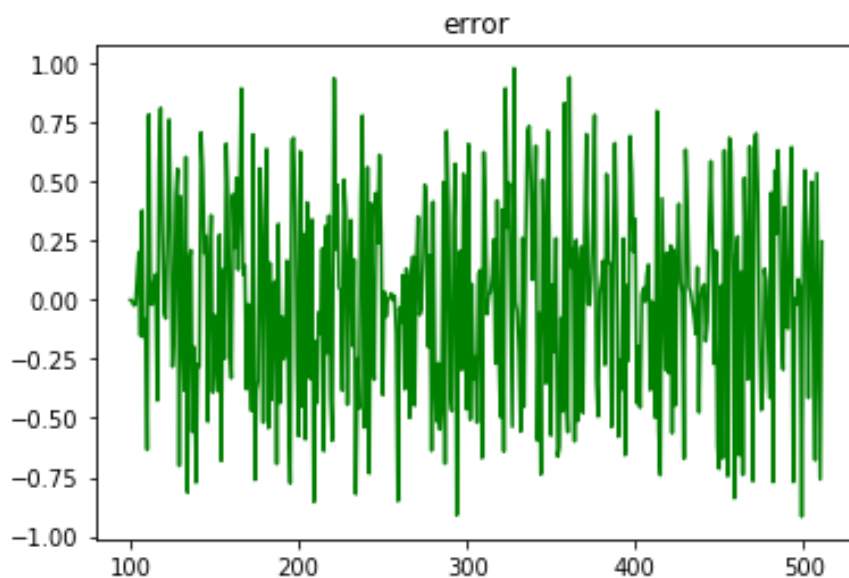


Figura 32: Erro com precisão de 15 dígitos usando embedding multidimensional

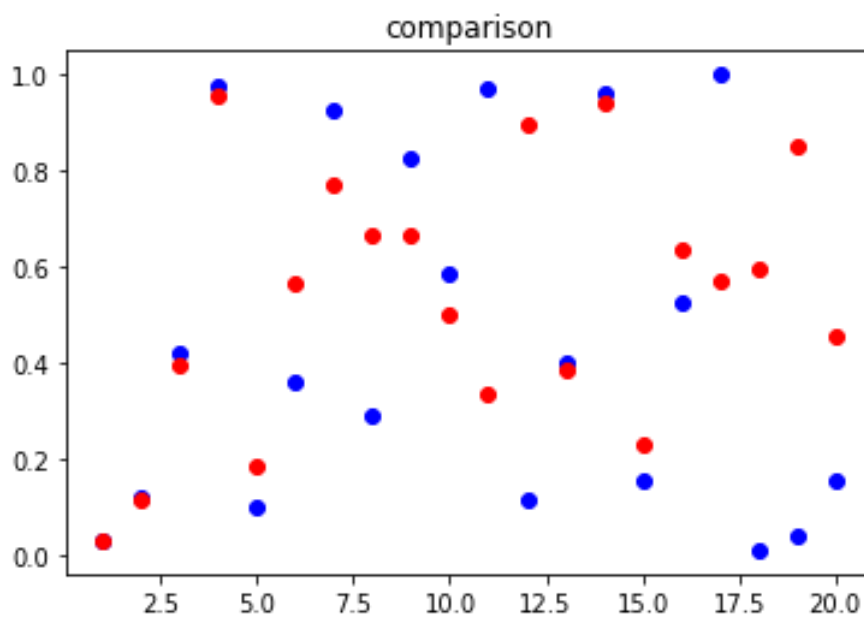


Figura 33: 20 primeiros pontos previstos para embedding multi dimensional e 15 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

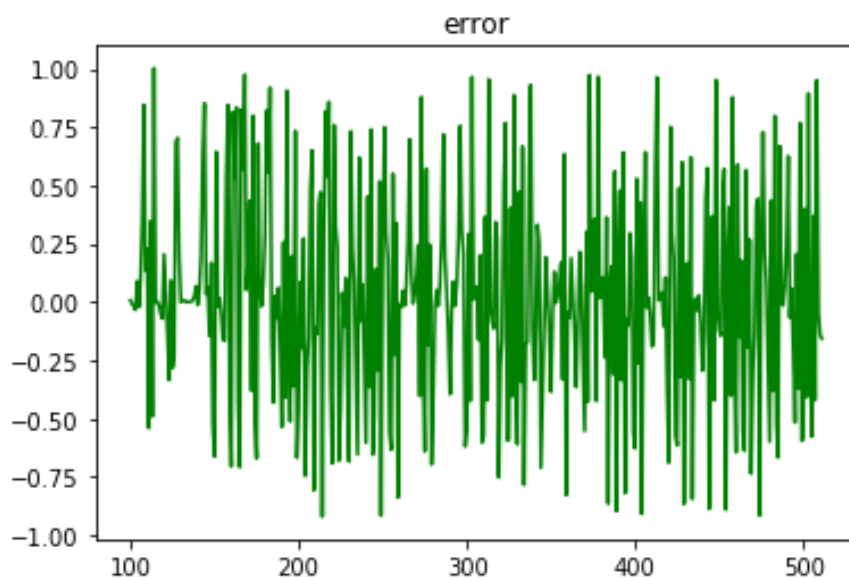


Figura 34: Erro com precisão de 14 dígitos usando embedding em 2D

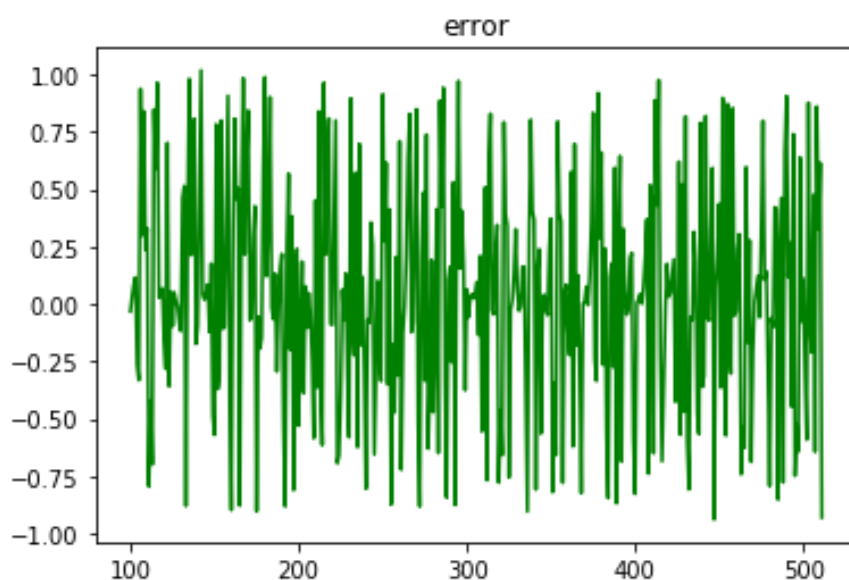


Figura 35: Erro com precisão de 14 dígitos usando embedding em 3D

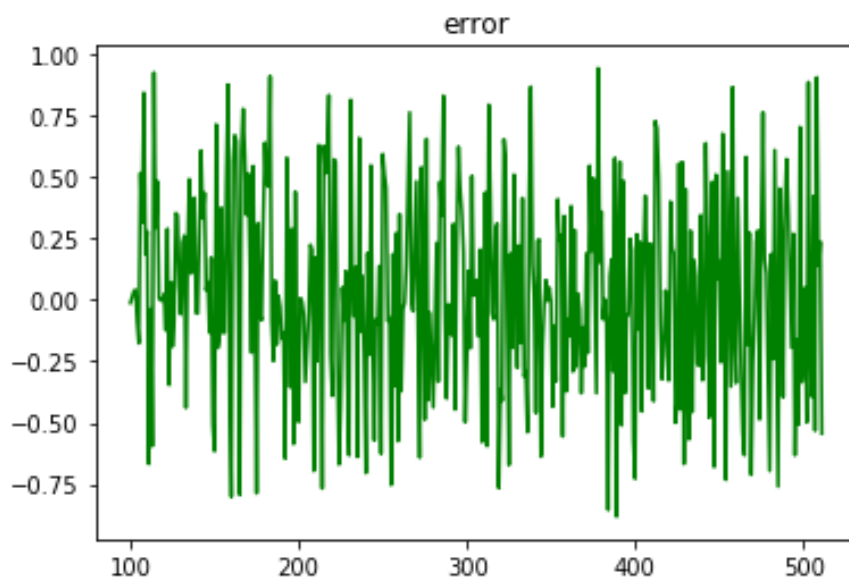


Figura 36: Erro com precisão de 14 dígitos usando embedding multidimensional

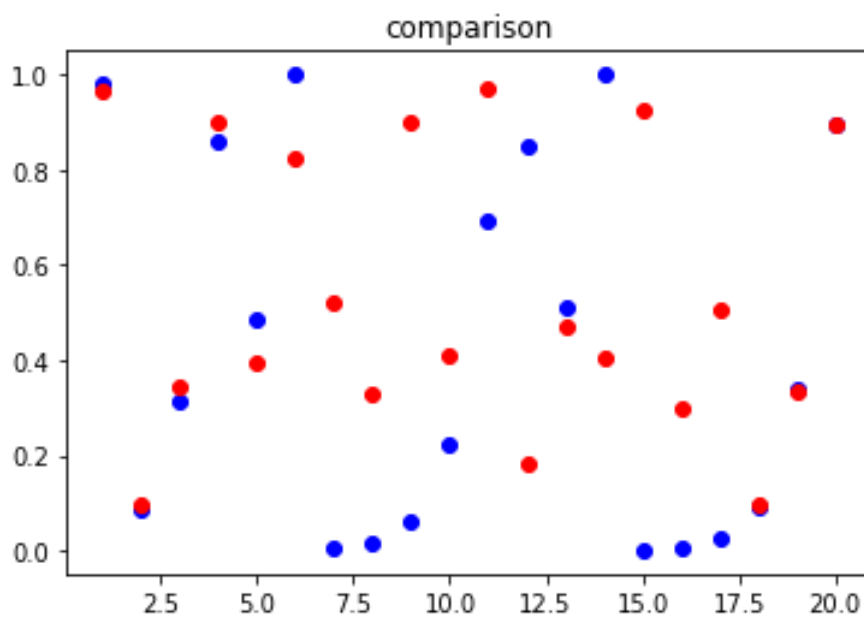


Figura 37: 20 primeiros pontos previstos para embedding multi dimensional e 14 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

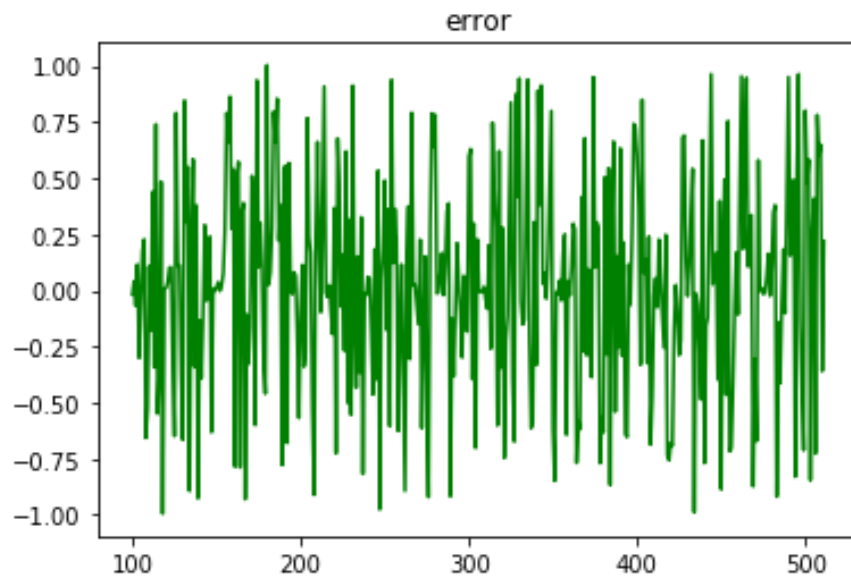


Figura 38: Erro com precisão de 13 dígitos usando embedding em 2D

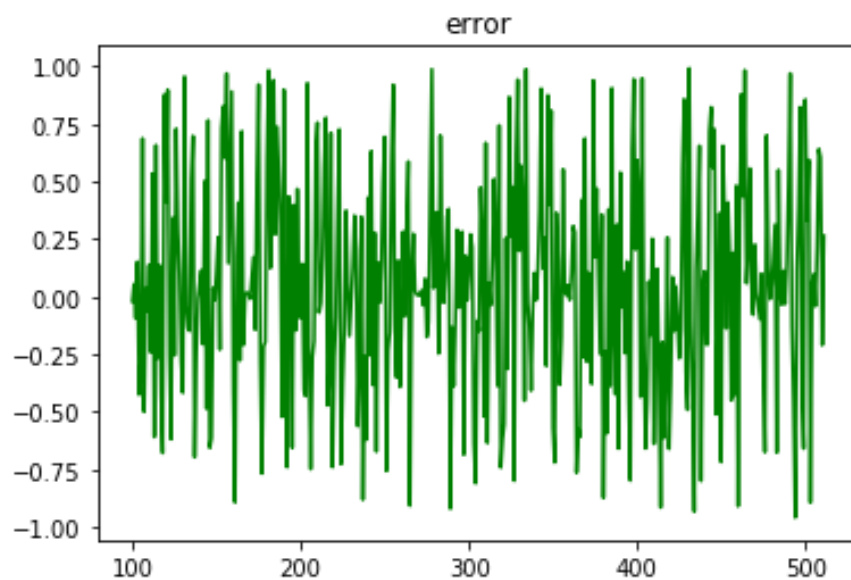


Figura 39: Erro com precisão de 13 dígitos usando embedding em 3D

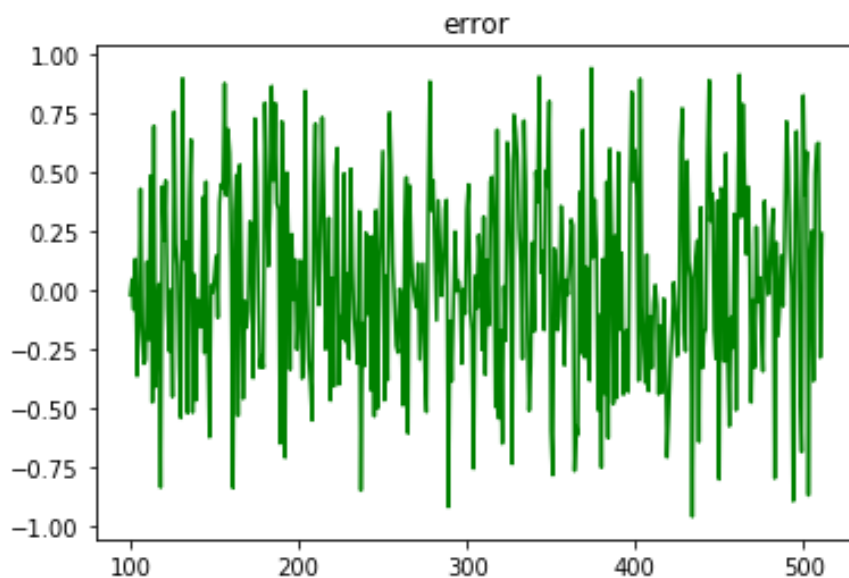


Figura 40: Erro com precisão de 13 dígitos usando embedding multidimensional

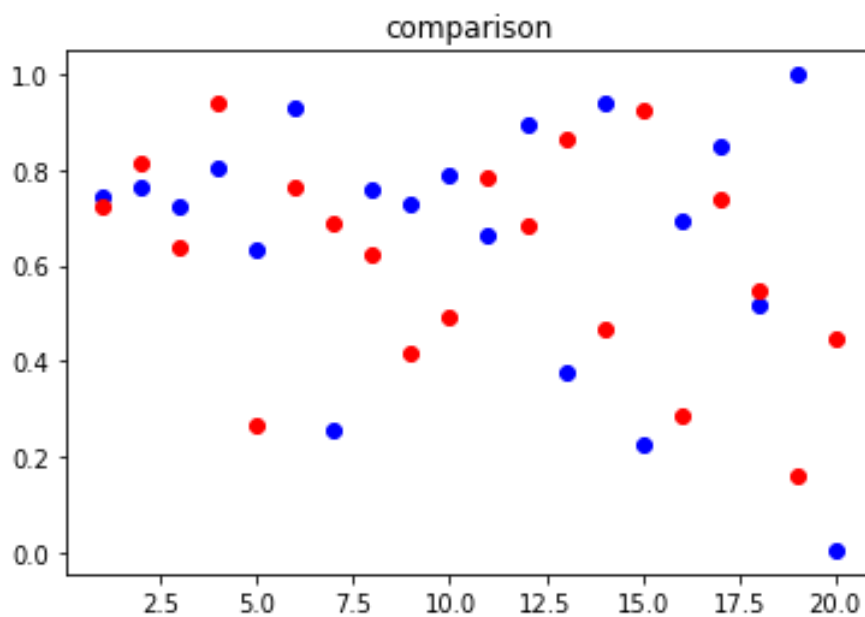


Figura 41: 20 primeiros pontos previstos para embedding multi dimensional e 13 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

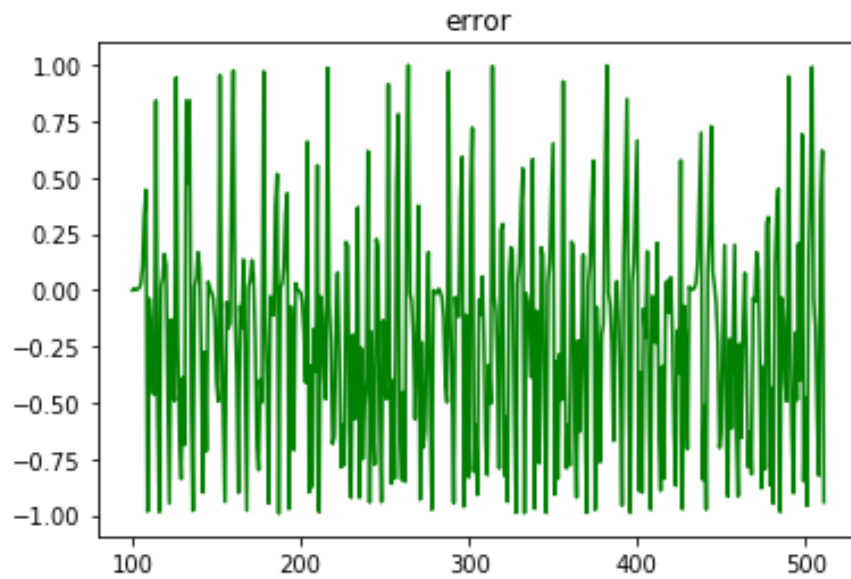


Figura 42: Erro com precisão de 12 dígitos usando embedding em 2D

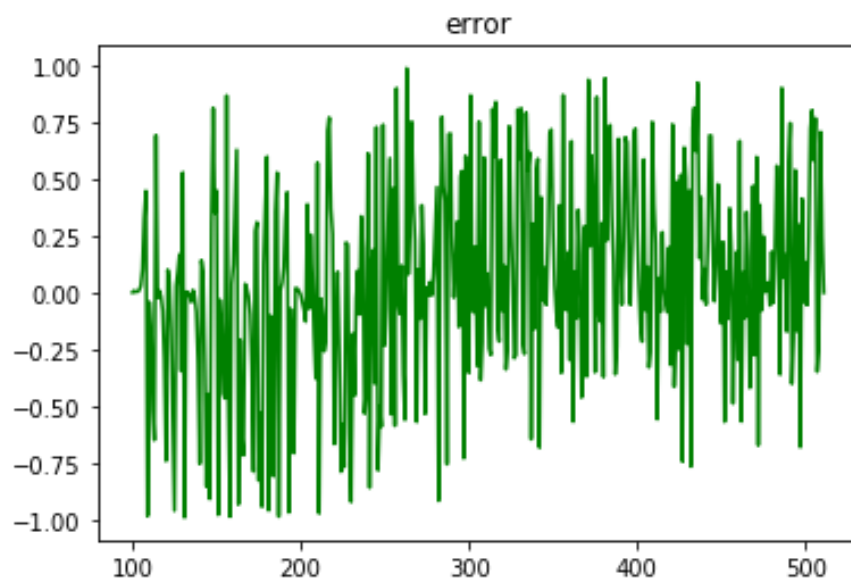


Figura 43: Erro com precisão de 12 dígitos usando embedding em 3D

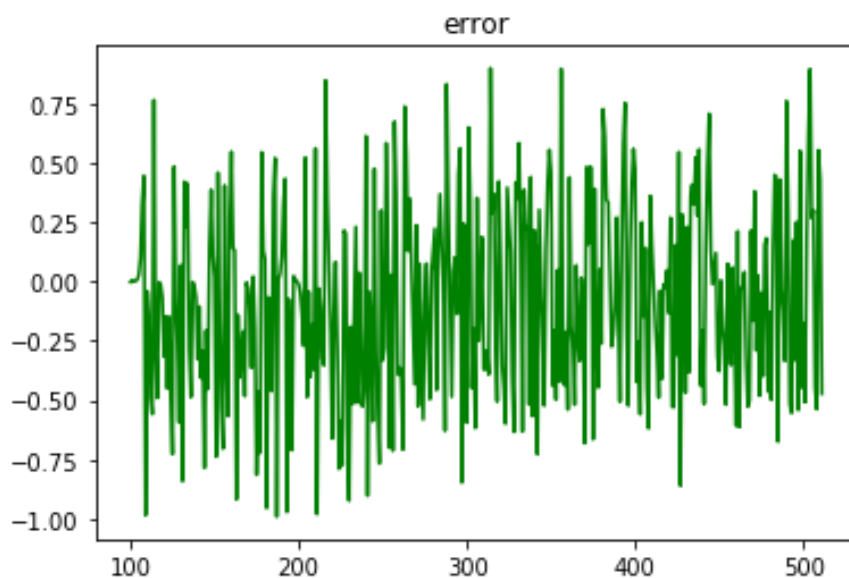


Figura 44: Erro com precisão de 12 dígitos usando embedding multidimensional

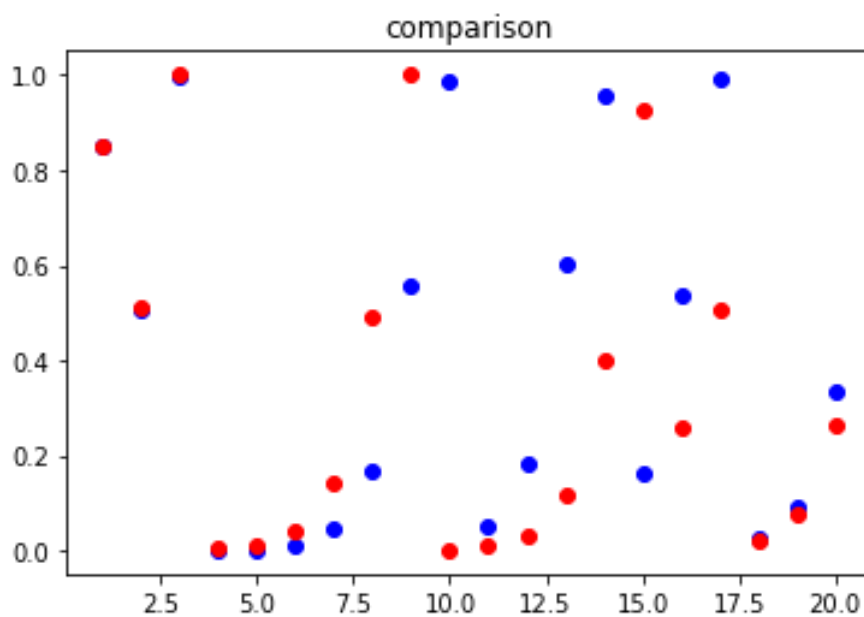


Figura 45: 20 primeiros pontos previstos para embedding multi dimensional e 12 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

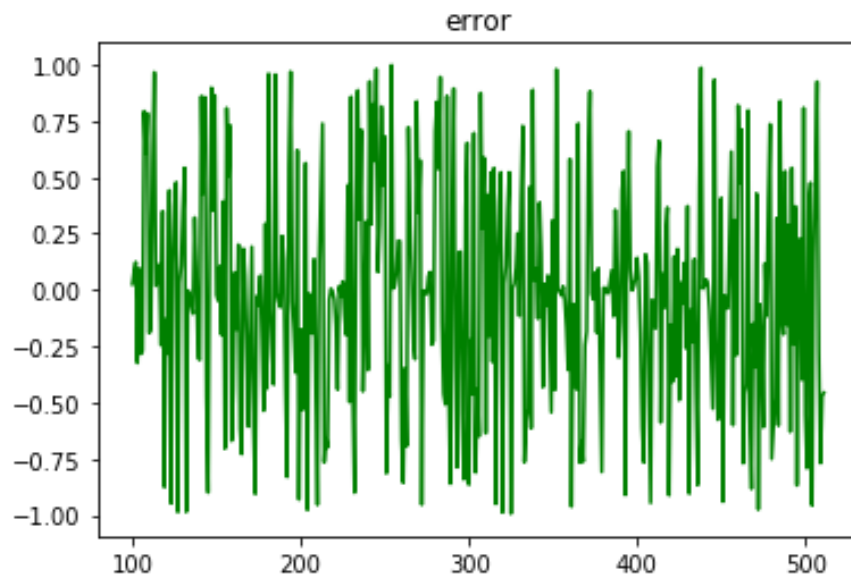


Figura 46: Erro com precisão de 11 dígitos usando embedding em 2D

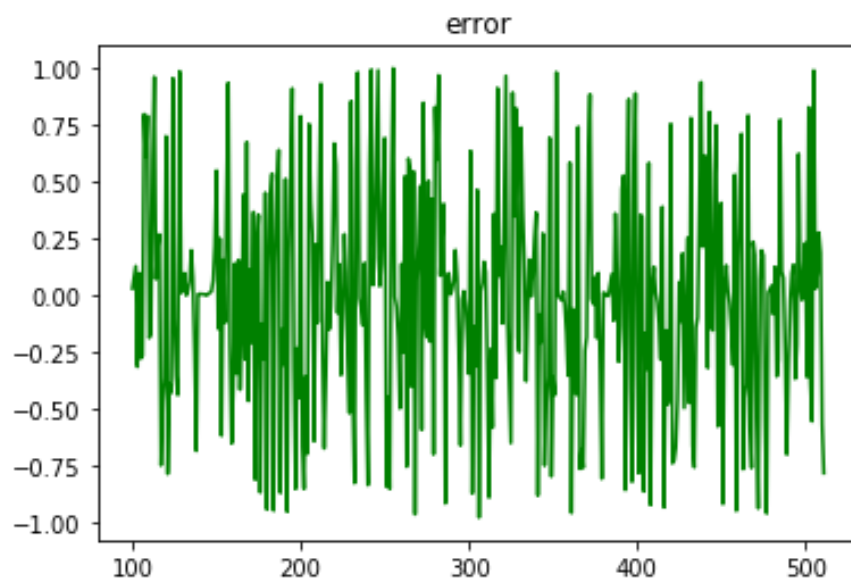


Figura 47: Erro com precisão de 11 dígitos usando embedding em 3D

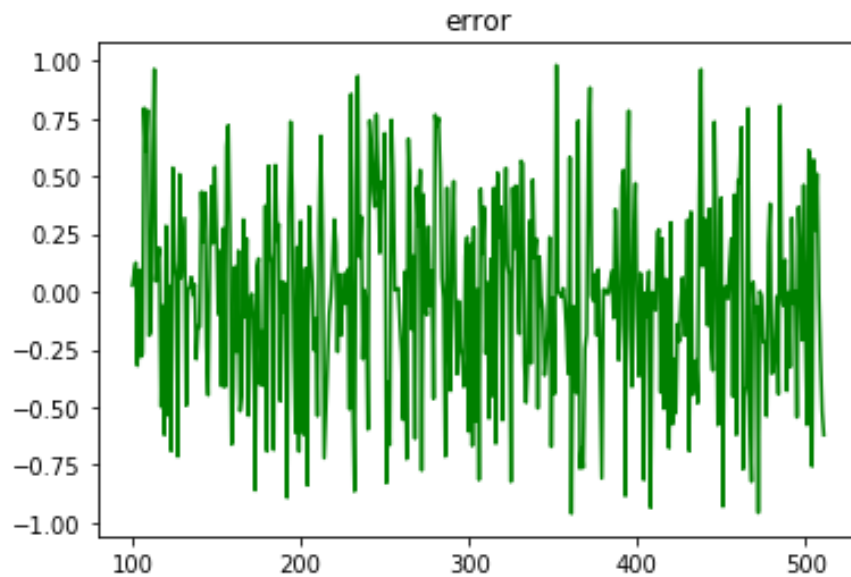


Figura 48: Erro com precisão de 11 dígitos usando embedding multidimensional

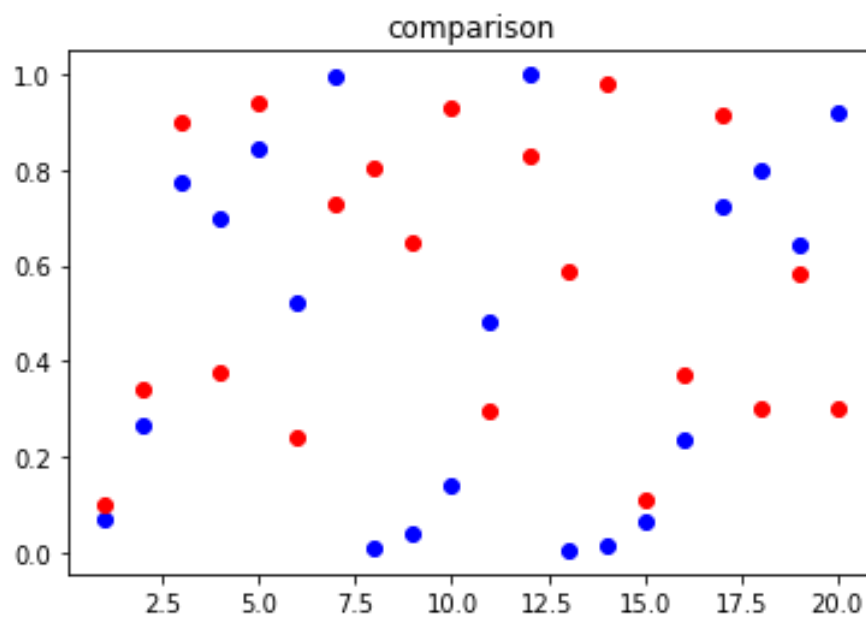


Figura 49: 20 primeiros pontos previstos para embedding multi dimensional e 11 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

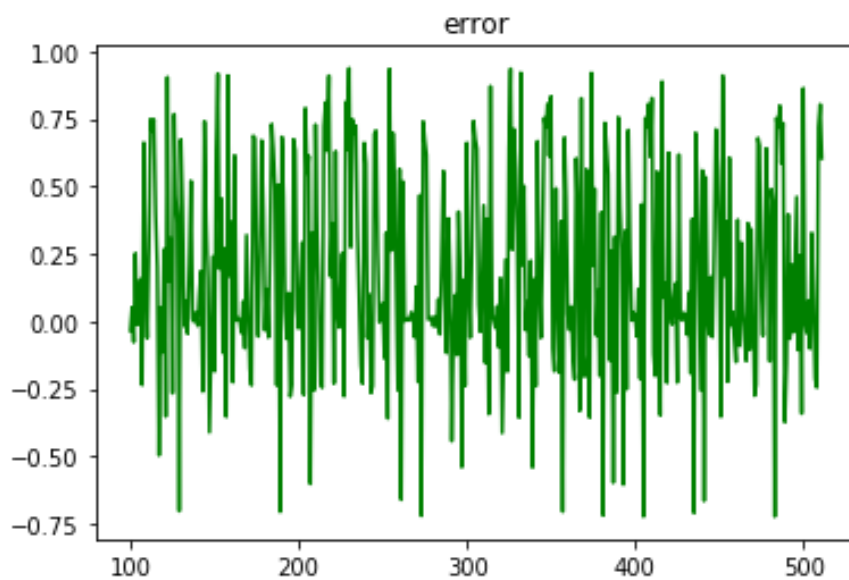


Figura 50: Erro com precisão de 10 dígitos usando embedding em 2D

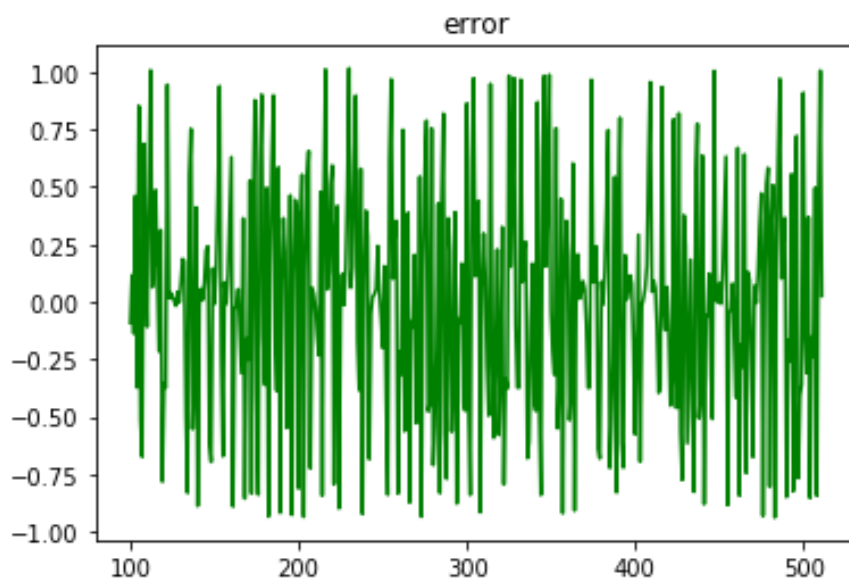


Figura 51: Erro com precisão de 10 dígitos usando embedding em 3D

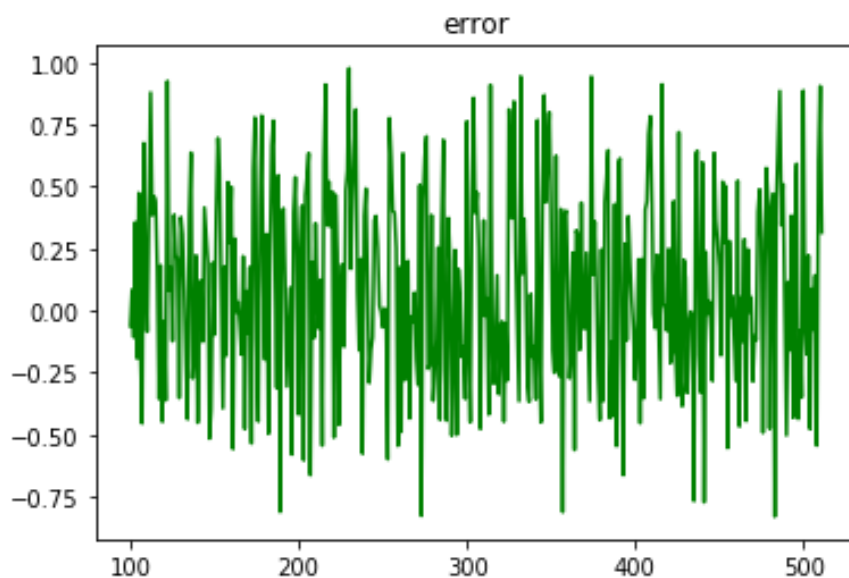


Figura 52: Erro com precisão de 10 dígitos usando embedding multidimensional

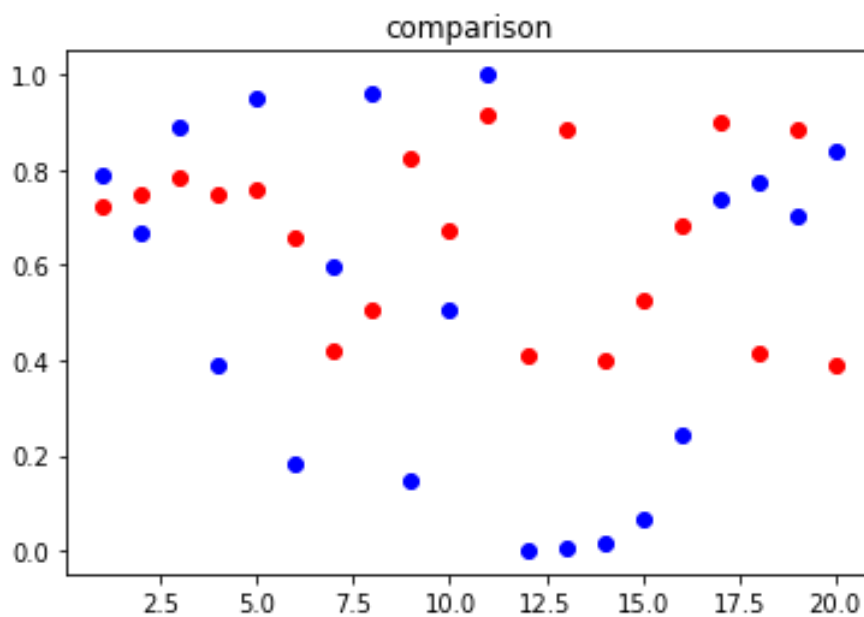


Figura 53: 20 primeiros pontos previstos para embedding multi dimensional e 10 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

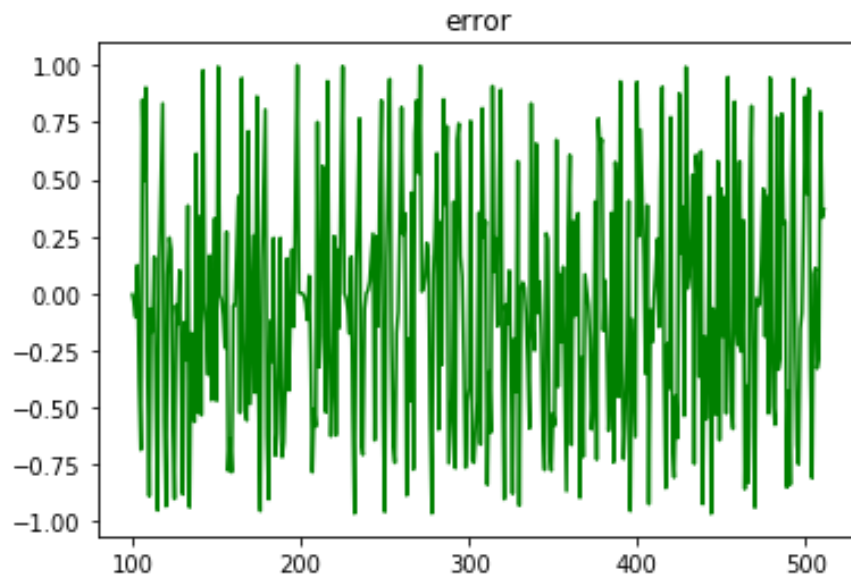


Figura 54: Erro com precisão de 9 dígitos usando embedding em 2D

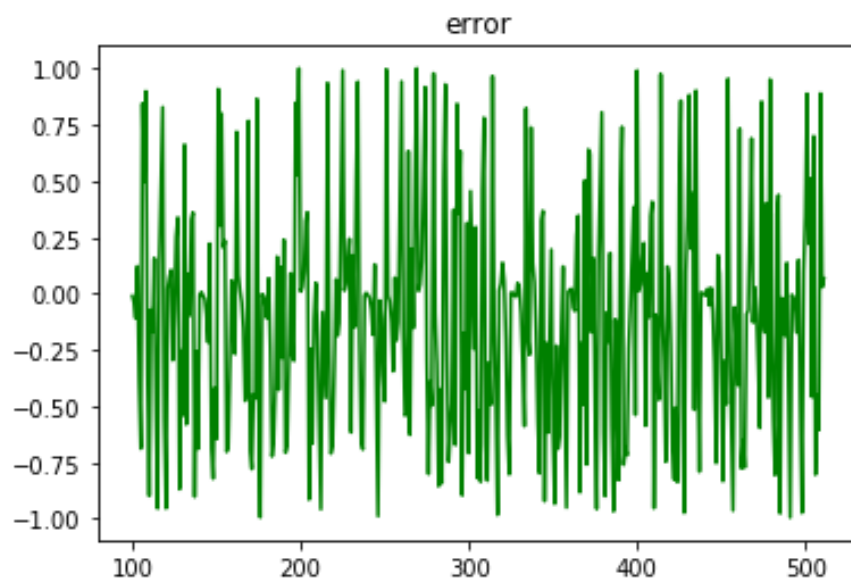


Figura 55: Erro com precisão de 9 dígitos usando embedding em 3D

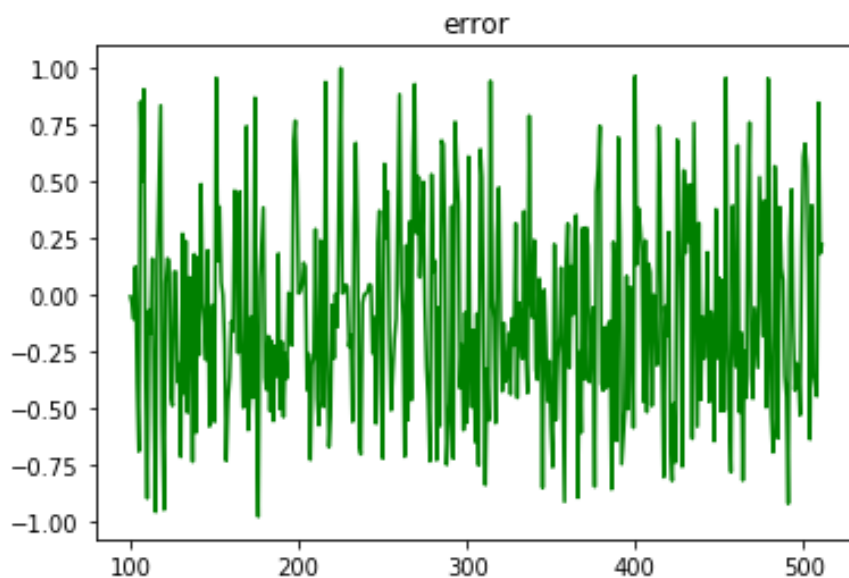


Figura 56: Erro com precisão de 9 dígitos usando embedding multidimensional

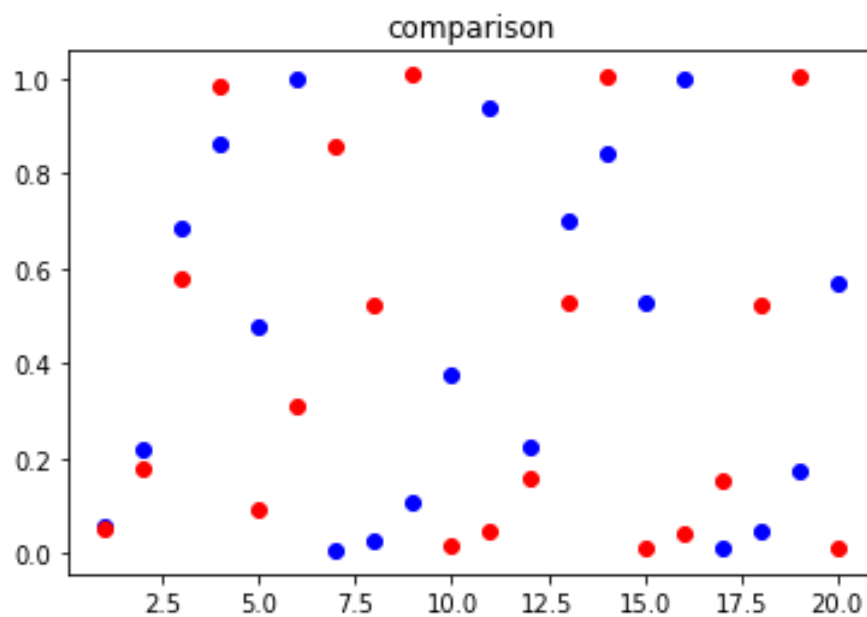


Figura 57: 20 primeiros pontos previstos para embedding multi dimensional e 9 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

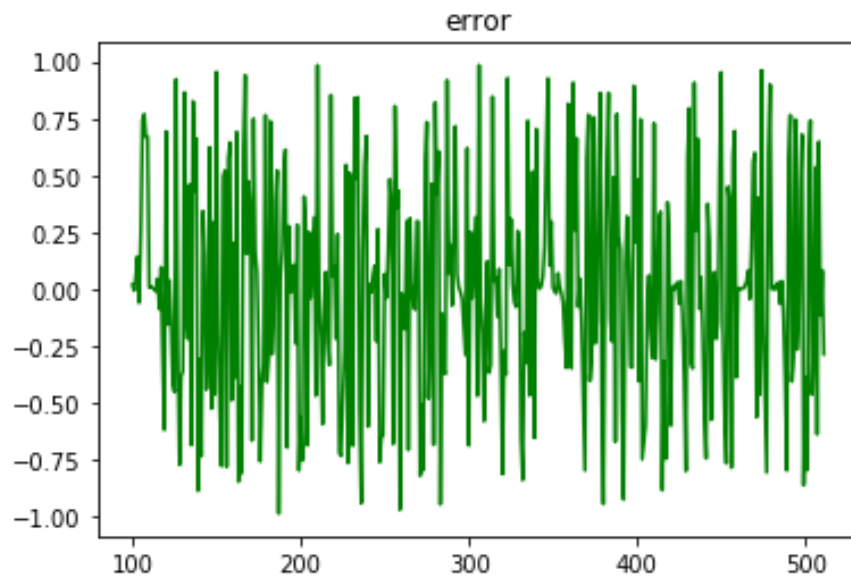


Figura 58: Erro com precisão de 8 dígitos usando embedding em 2D

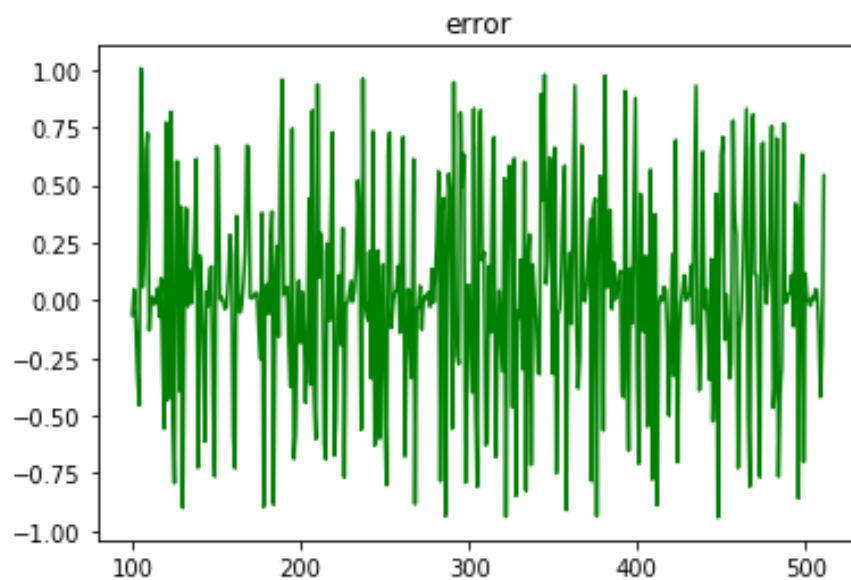


Figura 59: Erro com precisão de 8 dígitos usando embedding em 3D

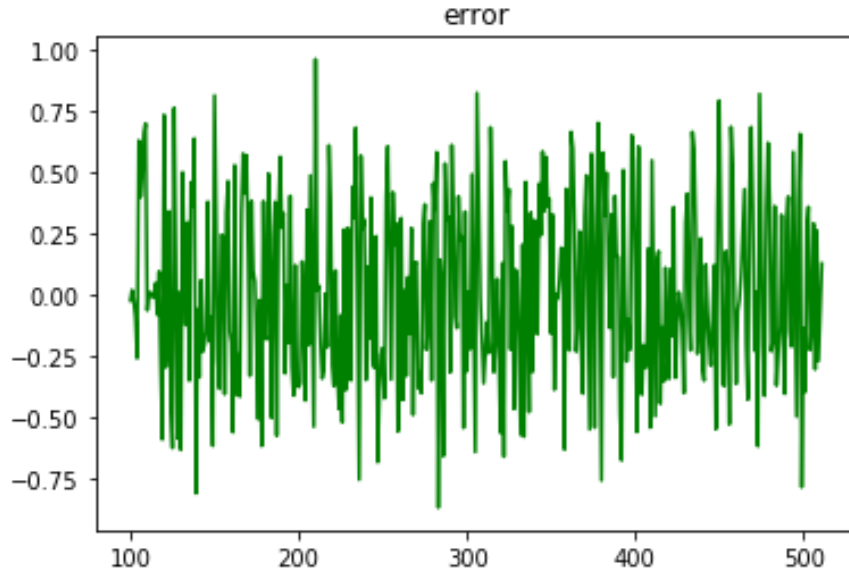


Figura 60: Erro com precisão de 8 dígitos usando embedding multidimensional

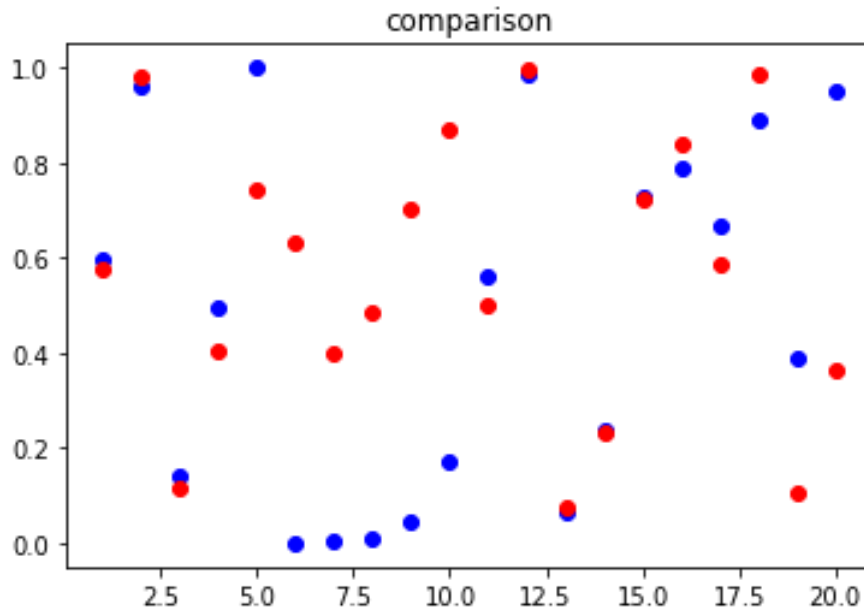


Figura 61: 20 primeiros pontos previstos para embedding multi dimensional e 8 dígitos. Pontos em azul são os reais e os em vermelhos são os obtidos pelo método

3.2 Algumas observações

Os resultados obtidos com a proposta 2 são superiores a proposta 1, além de não apresentarem o problema da exponencial em alguns embeddings 3D, no entanto, os resultados não estão como os descritos por Zhang, mesmo quando a precisão é alterada.