PREDIÇÃO DE SÉRIES TEMPORAIS EM INTERNET DAS COISAS COM REDES NEURAIS RECORRENTES

Samuel Silva de Oliveira
Departamento de Ciências Exatas e Tecnológicas - UNIFAP
Macapá / AP - Brasil
samuel.oliveira@unifap.br

Janine Kniess
Programa de Pós-Graduação em Computação Aplicada - UDESC
Joinville / SC - Brasil
janine.kniess@udesc.br

Rafael Stubs Parpinelli
Programa de Pós-Graduação em Computação Aplicada - UDESC
Joinville / SC - Brasil
rafael.parpinelli@udesc.br

William Alberto Cruz Castañeda
Departamento de Informática - UTFPR
Guarapuava / PR - Brasil
wcastaneda@utfpr.edu.br

RESUMO

A popularização da Internet das Coisas resultou na implantação de diversos tipos de sensores em diferentes áreas, como monitoramento do ambiente, indústrias, energia, e outros. Esse grande volume de dispositivos que estão conectados à internet, gera quantidades massivas de dados a serem analisados. Muitos desses dados são ordenados pelo tempo, tais como Séries Temporais. Este artigo apresenta um modelo de predição de séries temporais baseado em Redes Neurais Recorrentes que realiza a predição de um passo à frente. Foram utilizados cinco conjuntos de dados de sensores de Internet das Coisas para a realização dos experimentos. Os resultados obtidos demonstraram que a abordagem utilizada obteve melhor desempenho que o modelo ARIMA e SVM.

PALAVRAS CHAVE. Redes Neurais Recorrentes, Internet das Coisas, Séries Temporais.

Tópicos: MH - Metaheurísticas

ABSTRACT

Internet of Things emerged in such a way that resulted in the deployment of several types of sensors in various application fields, such as environment monitoring, industries, energy and others. This great volume of connected devices creates massive quantities of data to be analyzed. In many cases, this data are ordered by time, just like a Time Series. This paper presents a model of time series prediction based on Recurrent Neural Networks in order to predict one step ahead. It was used five *datasets* from IoT sensors in the tests. The performed tests showed that the Recurrent Neural Network obtained better performance than ARIMA and SVM.

KEYWORDS. Recurrent Neural Networks. Internet of Things. Time-series.

Paper topics : MH - Metaheuristics

https://proceedings.science/p/85353?lang=pt-br

1. Introdução

Nos últimos anos, sensores de Internet das Coisas[1] têm sido massivamente implantados em diversos cenários de aplicação visando monitorar valores de variáveis físicas oferecendo suporte à decisão. Segundo [Gartner, 2017], no ano de 2017, mais de 8 bilhões de dispositivos de Internet das Coisas estariam em funcionamento, o que representa um acréscimo de 31% com relação ao ano anterior. É certo que a quantidade desses dispositivos em funcionamento é crescente, implicando em um grande volume de dados sendo criado continuamente. [Kinzler, 2017] ressalta que é necessário que existam sistemas cognitivos capazes de aprender e interagir naturalmente com humanos para obter informações sobre os dados coletados e ajudá-los a tomar melhores decisões, aumentando a importância e o valor desses dados.

Em grande parte dos casos, os dados de sensores são ordenados pelo tempo, formatados como Séries Temporais.

De acordo com [Morettin e de Castro Toloi, 1981], uma série temporal pode ser definida como sendo qualquer conjunto de observações ordenadas no tempo. Tais observações podem demonstrar vários comportamentos, como os listados abaixo:

- **Tendência** - Elementos de longo prazo relacionados com a série temporal;
- **Ciclo** - Ondas relativamente regulares, em torno de uma linha de tendência;
- **Sazonalidade** Padrões regulares da série de tempo;
- **Aleatoriedade** - Efeitos que não foram incorporados pela série de tempo via os três componentes anteriormente citados, ou seja, é o resíduo.

A análise de séries temporais não está apenas relacionada com a identificação do comportamento da série, mas também com a predição de valores futuros. Tal funcionalidade é de grande importância em determinados contextos, como medição do nível de reservatórios, terremotos, qualidade do ar, dentre outros.

Atualmente, podemos encontrar vários modelos de predição de séries temporais, desde os mais simples e de fácil implementação, porém com pouca análise dos dados, até os mais complexos, baseados em formalidades matemáticas e estatísticas envolvendo um grande esforço na predição de valores futuros. Pode-se encontrar na literatura abordagens para predição de séries temporais baseadas em Algoritmos bio-inspirados, Machine Learning, Big Data, Deep Learning, Redes Neurais, e outros. Como a predição não é uma tarefa trivial, busca-se aliar algoritmos poderosos, como os mencionados acima, com grandes capacidades de processamento de dados, a fim de alcançar bons resultados com o menor tempo de resposta possível.

Este artigo apresenta como contribuição o desenvolvimento de um modelo de predição de séries temporais baseado em Redes Neurais Recorrentes aliado a técnicas de aprendizagem, como o Gradiente Descendente Estocástico, oferecendo melhor resultado que algoritmos clássicos como Autoregressive Integrated Moving Average (ARIMA) [Wu e Wei, 1989] e Support Vector Machine (SVM) [Smola e Schölkopf, 2004].

Redes Neurais Recorrentes diferem das Redes Neurais tradicionais (*feedfoward*) pelo fato de terem a capacidade de receber sinais tanto da camada de entrada quanto da camada oculta na iteração de tempo anterior [Sundermeyer et al., 2015]. De certa forma, a camada oculta simula o funcionamento de uma memória. Além disso, as Redes Neurais Recorrentes trabalham com dados sequenciais tanto na camada de entrada quanto na camada de saída, se encaixando perfeitamente no contexto de séries de tempo. Testes comparativos também são apresentados neste estudo, comparando o modelo proposto com outros modelos clássicos encontrados na literatura que receberam bastante atenção nos últimos anos: ARIMA e SVM.

[1] Do Inglês: Internet of Things - IoT

https://proceedings.science/p/85353?lang=pt-br

O restante deste trabalho está dividido da seguinte forma: Na Seção 2 apresentam-se os trabalhos relacionados; Na Seção 3 demonstra-se o modelo de predição de séries temporais proposto neste estudo; Na Seção 4 apresentam-se os resultados obtidos e análises, e por fim, na Seção 6, as conclusões e trabalhos futuros.

2. Trabalhos relacionados

Nesta seção serão discutidos alguns trabalhos encontrados na literatura que lidam com o problema da predição de séries temporais baseando-se em Redes Neurais.

[George e Mutalik, 2017] propôs uma solução baseada na utilização de redes neurais *feedfoward* de uma única camada juntamente com um Algoritmo de Aprendizagem Sequencial Online (OSLA). Os autores apresentam uma perspectiva matemática que visa mapear os valores presentes e valores prévios através de um Sistema Não-Linear Geral. Comparações com outros modelos são apresentadas utilizando vários conjuntos de dados (*datasets*) e os resultados mostram que a utilização do OSLA obteve melhor desempenho que os outros modelos comparados.

[Xu et al., 2016] definiu um modelo específico para predição de séries temporais multivariadas utilizando Redes Neurais Hierárquicas aliadas a reservatórios de ciclo simples (SCR) e aprendizado de máquina. O algoritmo *Particle swarm optimization* (PSO) é utilizado para o estágio de treino a fim de otimizar parâmetros dos SCR's. Os resultados da simulação demonstram que o modelo proposto supera outros modelos utilizados na comparação. Sua estrutura hierárquica é especialmente projetada para capturar as características dinâmicas e complexas das séries temporais multivariadas.

[Hussein et al., 2016] introduziu um método baseado em Redes Neurais Recorrentes Coevolucionárias para a predição de séries temporais caóticas com o foco em predições de muitos passos à frente. Para o treinamento da rede neural, foram utilizadas as técnicas de *Backpropagation Through Time* (BPTT) e *Cooperative Neuro-Evolution* (CNE). Para a realização dos testes, foram utilizados *datasets* sintéticos e reais. Os resultados mostraram que a predição de muitos passos à frente ainda é um grande desafio, pois o acúmulo dos erros nas predições resultam em resultados finais com grandes erros.

[Guo et al., 2016] apresentou um método de aprendizagem de gradiente adaptativo para Redes Neurais Recorrentes a fim de prever séries temporais com anomalias. O algoritmo proposto visa aprender incrementalmente as séries temporais e fornecer previsões robustas adaptando-se aos padrões em mudança, bem como a resistir a *outliers*[2]. Experimentos demonstraram que a abordagem proposta apresenta superioridade com relação à outros métodos.

Embora os estudos mencionados acima demonstrem bons resultados, as implementações visam objetivos bem específicos além de serem bastante complexas. Este estudo visa propor um modelo simples e de fácil implementação para a predição de qualquer tipo de série temporal gerada por dispositivos de Internet das Coisas.

3. Modelo de Predição de Séries Temporais com Redes Neurais Recorrentes

Como dito anteriormente, as Redes Neurais Recorrentes têm a capacidade de receber dados da camada de entrada e mantê-los para serem utilizados na próxima iteração. Desta forma, o fluxo da informação não segue em um único sentido, nem o resultado final depende exclusivamente da entrada atual, mas também das entradas anteriores. Com a meta de investigar a aplicação de Redes Neurais Recorrentes em dados coletados de sensores atuando em aplicações de IoT, esta seção descreve o modelo de predição de séries temporais com Redes Neurais Recorrentes e os métodos para realizar predições um passo à frente. É importante ressaltar que o modelo proposto neste estudo pode lidar com qualquer série temporal. O foco em Internet das Coisas se deu por conta da variedade de comportamentos que os *datasets* de IoT podem demonstrar. Ao invés de implementar um modelo específico para cada comportamento (ciclo, tendência, aleatoriedade e outros), o modelo baseado na Rede Neural Recorrente pode fazer boas previsões, independente do seu comportamento.

[2]Observações que apresentam grande afastamento das demais. Valores atípicos.

https://proceedings.science/p/85353?lang=pt-br

3.1. Redes Neurais Recorrentes

[Barreto, 2002] explica que o estado oculto da Rede Neural Recorrente recebe informações das variáveis independentes, assim como de seu próprio resultado de processamento da iteração anterior, como ilustrado na Figura 1.

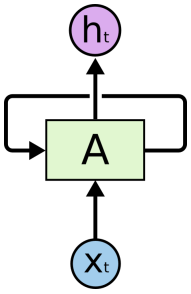


Figura 1: Neurônio da camada Oculta da Rede Neural Recorrente [Olah, 2014]

Neste exemplo, X_t é representado pelo dado de entrada na iteração de tempo t, o qual será processado pelo neurônio A. h_t é representado pela saída do neurônio A na iteração de tempo t, que pode ser utilizada nas próximas iterações pelo mesmo neurônio, como comportamento padrão de recorrência.

O efeito prático disto é similar ao comportamento de uma memória de curto prazo e é bastante útil no contexto de predição de séries temporais, tendo em vista que os dados das camadas de entrada e saída são dados sequenciais [Goodfellow et al., 2016]. Redes Neurais geralmente são utilizadas para regressão e classificação. O modelo proposto neste artigo utiliza Redes Neurais Recorrentes para regressão, que pode ser entendida como uma equação para se estimar o valor esperado de uma variável y, dados os valores de algumas outras variáveis x.

3.2. TensorFlow

Como o foco deste estudo é prover um modelo de fácil implementação, o TensorFlow [Abadi et al., 2016] foi utilizado. O TensorFlow é um projeto do Google focado na implementação e implantação de modelos de Aprendizagem de Máquina em larga escala. Sua abordagem é baseada em um modelo flexível de programação baseada em fluxo de dados aplicado a diversas plataformas de hardware, sejam elas centralizadas ou distribuídas. É possível utilizá-lo ainda para expressar uma grande variedade de algoritmos, incluindo algoritmos de treinamento para Redes Neurais de forma escalável. A Interface de Programação de Aplicativos (API) principal é na linguagem de programação Python, com implementação de computações extremamente eficientes também em C++ [Abadi et al., 2016].

O TensorFlow suporta o processamento em CPU's, tanto quanto em GPUs para maior paralelização das execuções. Neste estudo, a linguagem de programação utilizada foi Python, juntamente com outras bibliotecas existentes para a manipulação dos dados, como Pandas[3], NumPy[4] e MatPlotLib[5].

3.3. Funções Custo

Em aprendizado de máquina, existe um custo[6] envolvido no processo de aprendizado. Segundo [Goodfellow et al., 2016], esse custo é representado pelo tamanho do erro obtido de acordo com os valores dos parâmetros que foram utilizados. Efetivamente, custo é o tamanho do desvio que temos para um determinado conjunto de parâmetros. Neste caso, é necessário minimizar a função de custo para que a performance do aprendizado seja melhorada.

[3]https://pandas.pydata.org/
[4]http://www.numpy.org/
[5]https://matplotlib.org/
[6]Do Inglês: Loss

https://proceedings.science/p/85353?lang=pt-br

O método do Gradiente Descendente é utilizado em otimização com a finalidade de minimizar a função custo utilizando um esquema iterativo. Tendo em vista que a função em questão é convexa, a cada iteração se toma a direção negativa do gradiente até que o mínimo global seja alcançado. Porém, este método pode não ser ideal quando se tem um conjunto de dados muito grande, porque neste caso, o algoritmo precisaria percorrer o conjunto de dados por inteiro para computar o gradiente e dar apenas um passo na otimização.

Tendo em vista que os dados de Internet das Coisas são massivos, neste estudo foi utilizado o método de regressão linear por Gradiente Descendente Estocástico. Neste método são criados mini-lotes de dados para serem percorridos pelo algoritmo de forma aleatória. Consequentemente, não é necessário percorrer todos os dados para computar o gradiente e é possível obter uma aproximação apenas percorrendo alguns exemplos do conjunto de dados. É fato que para chegar mais próximo do mínimo, será necessária uma grande quantidade de iterações, porém, o tempo da iteração será consideravelmente curto, e o aprendizado mais rápido [Bottou, 2010].

3.4. Métricas para Avaliação

Identificou-se necessária a utilização de métricas para definir o nível de assertividade do modelo preditivo. Dentre as métricas mais utilizadas em aprendizado de máquina, destaca-se o Erro Absoluto Médio. O Erro Absoluto Médio mede o afastamento médio das previsões em relação aos valores observados, focando a estimativa na mediana condicional. [Willmott e Matsuura, 2005] realizaram um estudo demonstrando vantagens na utilização desta métrica sobre outras, como erro quadrático médio.

Outra métrica utilizada neste estudo é o coeficiente de determinação R^2. O coeficiente de determinação, é uma medida de ajustamento de um modelo estatístico linear generalizado, como a regressão linear, em relação aos valores observados. O valor de R^2 varia entre 0 e 1, indicando, em percentagem, o quanto o modelo consegue explicar os valores observados [Nagelkerke, 1991].

3.5. Conjuntos de Dados (*Datasets*)

Neste estudo, cinco conjuntos de dados de Internet das Coisas foram utilizados. São eles:

1. Dados de sensores de qualidade do ar - Universidade da Califórnia em Irvine [Vito et al., 2008];

   Este dataset está disponível no sítio da UCI Machine Learning Repository[7] que contém uma coleção de bases de dados disponibilizadas pela Universidade da Califórnia em Irvine para a comunidade de *Machine Learning* para análises empíricas de algoritmos de *Machine Learning*.

2. Dados do medidor inteligente de energia elétrica (SmartMeter) - UKPN [Networks, 2015];

   O projeto Low Carbon London disponibiliza dados de medidores inteligentes de 5.567 lares participantes em Londres - Reino Unido, no período de Novembro de 2011 a Fevereiro de 2015.

3. Dados ambientais de monitoramento da qualidade da água do Rio Burnett - Governo do Queensland (Austrália) [Queensland Government, 2015];

   Este dataset coleta séries temporais de leituras de parâmetros sobre a qualidade da água estauriana a cada meia hora no período de 2016 a 2018.

4. Níveis de Dióxido de Carbono Atmosférico - Laboratório de Pesquisa do Sistema Terrestre (Governo dos EUA) [Ed Dlugokencky, 2017] ;

   Neste conjunto de dados, constam níveis de emissão de CO2 a partir de quatro observatórios: Barrow, Alaska; Mauna Loa, Hawaii;, American Samoa e South Pole, Antártida.

[7] https://archive.ics.uci.edu/ml/datasets.html

5. Monitoramento de compressores de ar de cadeiras odontológicas [Castañeda, 2016];

   Cadeiras odontológicas contém compressores de ar. Frequentemente, parte do ar comprimido nos compressores sofre condensação, e consequentemente, os compressores ficam cheios de água, tornando as cadeiras inoperantes. Para evitar que as cadeiras fiquem fora de operação, sensores foram utilizados para monitorar a umidade nos compressores de cadeiras odontológicas a fim de expurgar a água acumulada em momento oportuno.

O objetivo principal deste estudo será prever um passo à frente dado o que pode ser observado nos últimos registros de cada *dataset*.

3.6. Etapas de Treino e Teste

Considerando o fato de que Redes Neurais Recorrentes recebem como entrada dados sequenciais, cada amostra dos dados precisa ser uma sequência, cujo formato é composto por períodos de tempo e variáveis. Assim, quando várias amostras são empilhadas para montar a base de dados, os mini-lotes terão um formato 3D: número de amostras, períodos de tempo e variáveis. Dessa forma, os dados poderão ser submetidos aos *sets* de treino e teste, sendo que cada mini-lote deve conter a defasagem de uma ou mais linhas para que os dados mantenham o padrão sequencial.

Como o objetivo do modelo se trata de previsão, não é ideal simplesmente separar aleatoriamente os dados para os *sets* de treino e teste. Para que a análise seja uma boa aproximação do cenário real, é preciso separar os últimos dados da série de tempo para servirem de *set* de teste. Neste estudo, 70% da série foi definida para treino e 30% para teste.

Uma vez que o método Gradiente Descendente Estocástico é utilizado no modelo proposto, é necessário embaralhar os mini-lotes de treino para que a função custo seja otimizada. Dessa forma é possível escolher de forma arbitrária o número de iterações para o *set* de treino. Nos testes realizados, identificou-se que a quantidade de iterações necessárias para a otimização do Erro Absoluto Médio varia de acordo com cada *dataset*, conforme ilustrado na Figura 2. Dessa forma, o parâmetro relacionado ao número de iterações do *set* de treino foi implementado de forma *online* e determinística, como explica [André e Parpinelli, 2014]. O algoritmo monitora o Erro Absoluto Médio e finaliza o set de treino caso o valor não diminua em uma determinada quantidade de iterações.

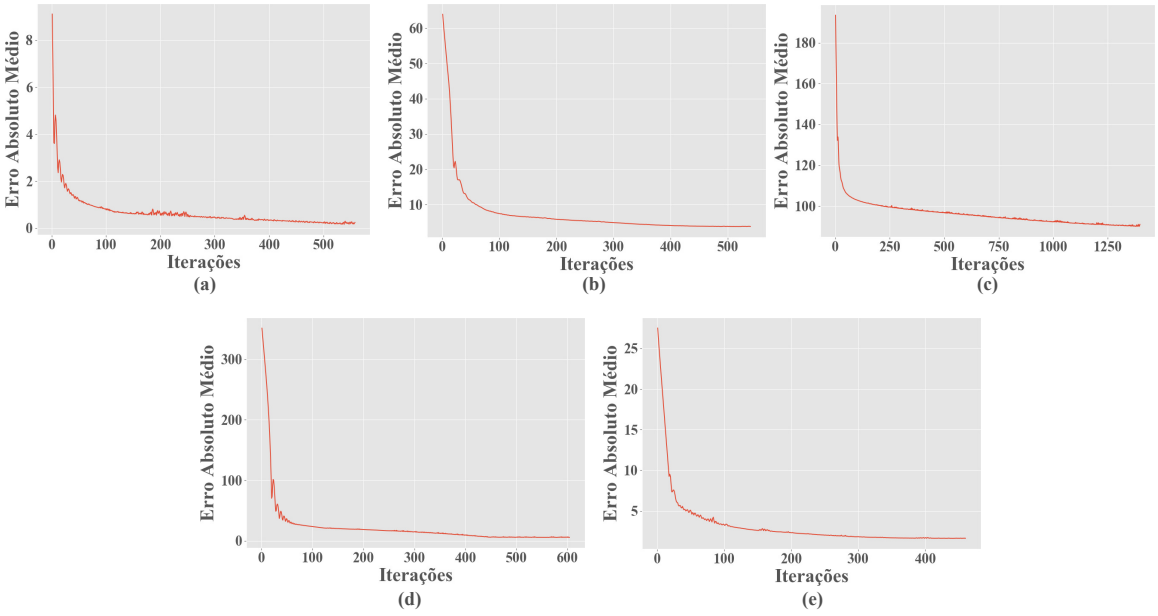


Figura 2: Otimização do Erro Absoluto Médio nas bases de Umidade (a), Qualidade do ar (b), SmartMeter (c), Emissão de CO2 (d) e Qualidade da Água (e)

https://proceedings.science/p/85353?lang=pt-br

Além disso, outros parâmetros da Rede Neural precisaram ser ajustados, como: número de neurônios, taxa de aprendizado, e tamanho dos mini-lotes. Tais parâmetros foram ajustados de forma *offline*, ou seja, antes da execução do algoritmo. A escolha dos valores desses parâmetros foi realizada de forma empírica, ou seja, baseada na observação do comportamento do algoritmo. São eles: Número de Neurônios: 64; *Learning Rate*: 0.001 e por fim, Tamanho dos mini-lotes: 64.

É importante ressaltar que outras séries temporais podem demonstrar comportamentos que exijam maior quantidade de iterações de treinamento para que a otimização do Erro Absoluto Médio chegue ao mais próximo possível do ótimo.

4. Resultados computacionais

Os resultados obtidos nas predições estão ilustrados nas Figuras 3, 4 e 5 desta seção. As linhas em vermelho representam os valores reais da série, e as amarelas representam os valores previstos. Apenas as primeiras 200 linhas do *set* de teste de cada *dataset* foram exibidas nas imagens a fim de obter melhor visualização dos resultados.

Com base na análise dos resultados, é possível identificar que o modelo de predição com Redes Neurais encontra dificuldades em lidar com os valores posicionados nas extremidades da série. Logo, as predições das séries de comportamento predominantemente sazonal podem não ter o melhor desempenho possível.

O modelo ARIMA, por sua vez, demonstra mais assertividade na predição dos valores posicionados nas extremidades, mas acaba acumulando erros ao longo da série, demonstrando um desempenho final inferior à Rede Neural.

Os diagramas de caixa apresentados na Figura 6 demonstram a variação de dados observados de Erro Absoluto Médio (EAM), Coeficiente de Determinação (R^2) e Tempo de execução (TE - em segundos) por meio de quartis. O Modelo de predição de séries temporais com ARIMA alcançou menor média de Erro Absoluto Médio, porém o modelo baseado na Rede Neural Recorrente alcançou maior média de Coeficiente de Determinação.

Na Tabela 1 são demonstrados os valores de Erro Absoluto Médio (EAM), Coeficiente de Determinação (R^2) e Tempo de execução (TE) obtidos na predição de cada conjunto de dados. Pode-se observar que na maioria dos casos a Rede Neural obtém melhores resultados.

Tabela 1: Resultados

	RNN	ARIMA	SVM
Smart Metter	EAM: 37.52	EAM: 51.81	EAM: 46.83
	R^2: 0.14	R^2: 0.05	R^2: 0.13
	TE (s): 2530.51	TE (s): 5.49	TE (s): 46.02
Burnett River	EAM: 5.963	EAM: 2.307	EAM: 8.708
	R^2: 0.96	R^2: 0.96	R^2: 0.46
	TE (s): 828.16	TE (s): 6.08	TE (s): 165.92
Compressor	EAM: 0.15	EAM: 1.01	EAM: 1.09
	R^2: 0.96	R^2: -1.01	R^2: -49.25
	TE (s): 1011.63	TE (s): 9.71	TE (s): 13.91
Emissão de CO2	EAM: 1.83	EAM: 2.11	EAM: 2.46
	R^2: 0.96	R^2: 0.98	R^2: 0.92
	TE (s): 77.77	TE (s): 3.10	TE (s): 9.23
Qualidade do Ar	EAM: 3.49	EAM: 5.07	EAM: 12.30
	R^2: 0.91	R^2: 0.76	R^2: -0.30
	TE (s): 101.8	TE (s): 2.96	TE (s): 7.77

A máquina utilizada para rodar os testes utiliza um processador AMD Phenom(tm) II X4 B93 Processor de 4 núcleos e 4Gb de memória RAM. Com relação ao tempo de processamento, o modelo ARIMA obteve maior velocidade. O tempo de execução é maior no caso da Rede Neural

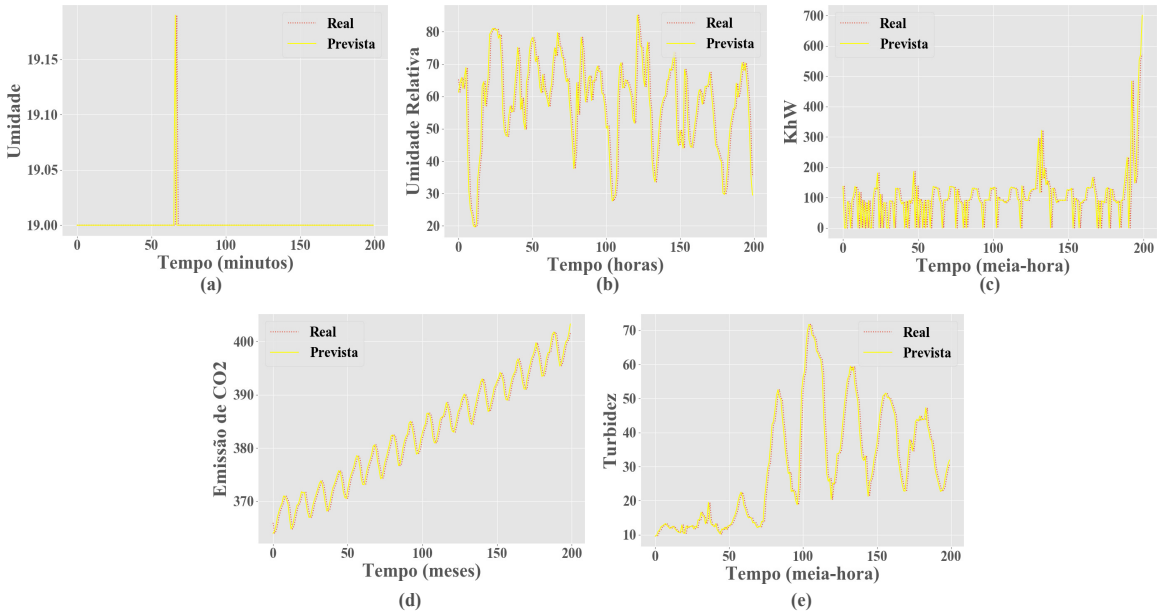


Figura 3: Predições das séries de Umidade (a), Qualidade do ar (b), SmartMeter(c), Emissão de CO2 (d) e Qualidade da Água (e) com Redes Neurais Recorrentes

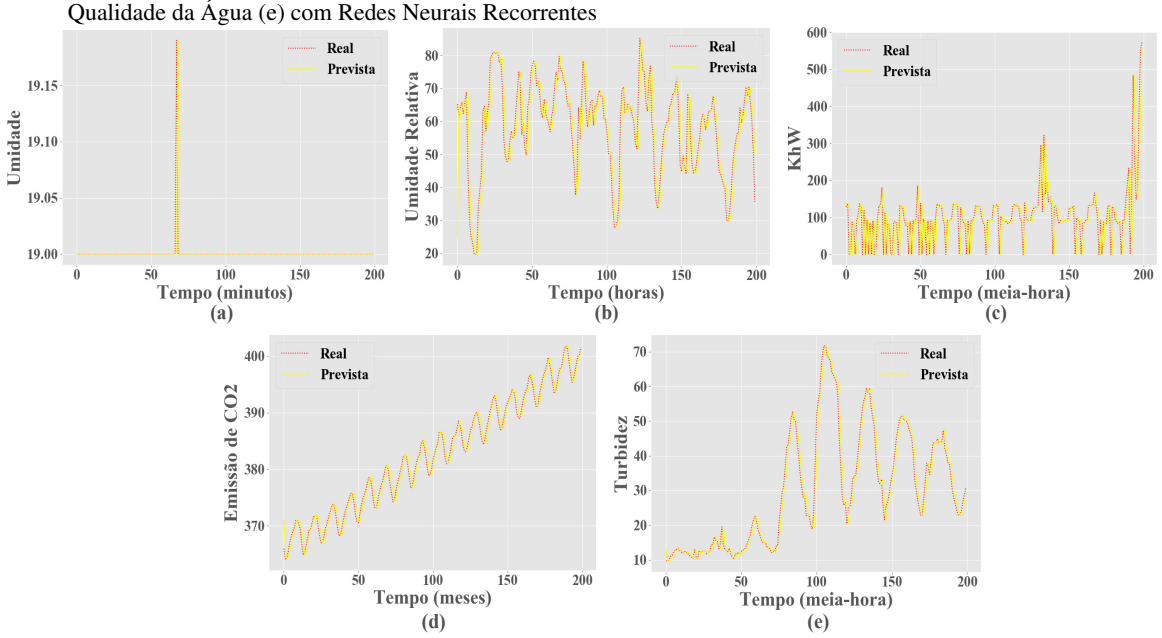


Figura 4: Predições das séries de Umidade (a), Qualidade do ar (b), SmartMeter(c), Emissão de CO2 (d) e Qualidade da Água (e) com ARIMA

Recorrente, pois compreende as fases de treino e teste. O SVM por sua vez, obteve o TE maior que o ARIMA. Identificou-se que o tempo que o algoritmo SVM levou até finalizar é diretamente proporcional ao tamanho do *set* de teste.

5. Conclusão

Este artigo apresentou um modelo de predição de séries temporais baseado em Redes Neurais Recorrentes implementada com o método Gradiente Descendente Estocástico a fim de minimizar o custo. As métricas utilizadas para a avaliação são: Erro Absoluto Médio (EAM) e Coeficiente de Determinação (R^2).

Os resultados mostraram que o modelo baseado na Rede Neural Recorrente apresentou melhor desempenho se comparado ao modelo ARIMA. O modelo baseado na Rede Neural Re-

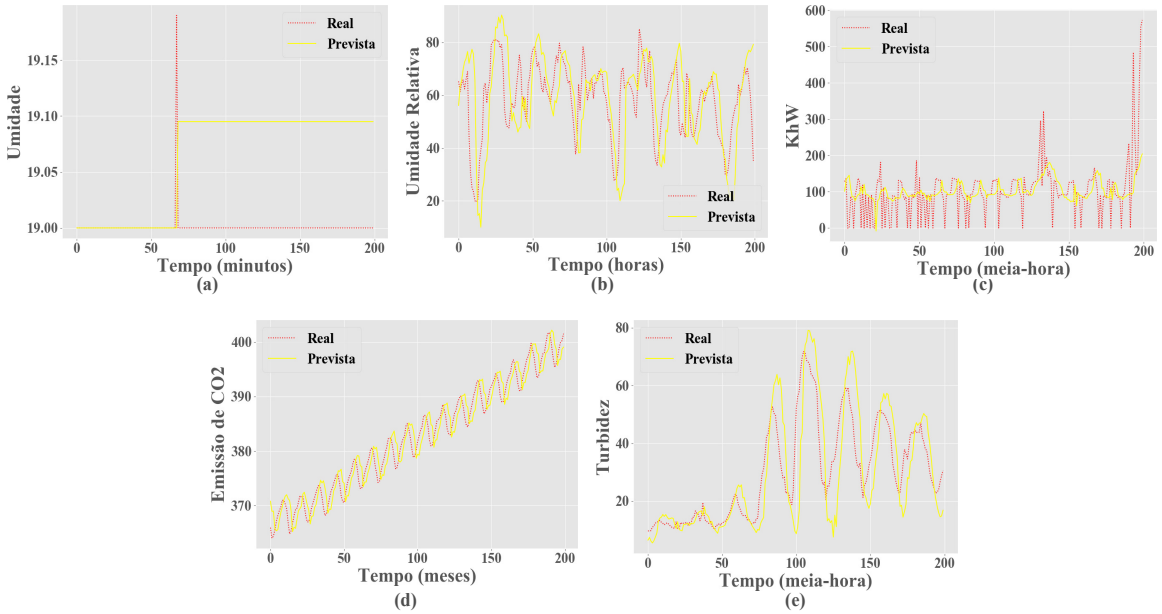


Figura 5: Predições das séries de Umidade (a), Qualidade do ar (b), SmartMeter(c), Emissão de CO2 (d) e Qualidade da Água (e) com SVM

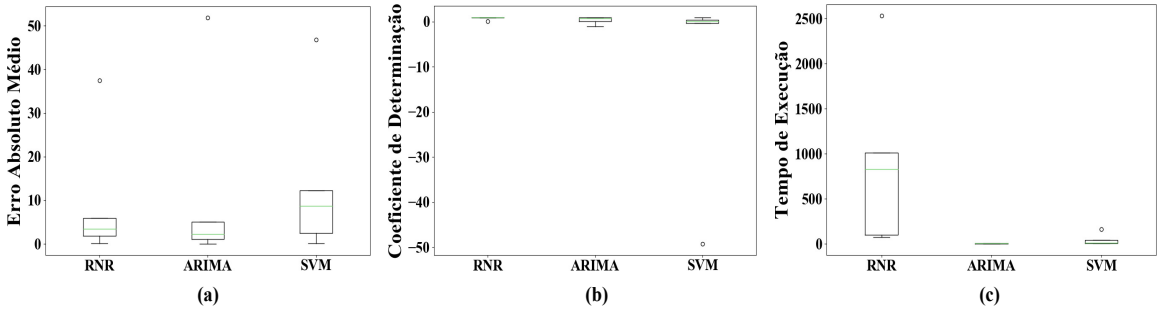


Figura 6: (a) Tempo de execução (TE), (b) Erro Absoluto Médio, (c) Coeficiente de Determinação (R^2)

corrente obteve ainda melhor desemepenho quando comparado ao modelo baseado em SVM, que alcançou maior Erro Absoluto Médio (EAM) e menor Coeficiente de Determinação (R^2).

Como perspectivas de trabalhos futuros, sugere-se melhorar o modelo de forma que possa utilizar a multi-variedade das séries temporais, visando obter previsões mais assertivas; Implementar a previsão para muitos passos à frente com a otimização do Erro Absoluto Médio e Coeficiente de Determinação; Aperfeiçoar o modelo proposto o tornando um modelo híbrido a fim de alcançar melhores resultados independente do comportamento da série, e por fim; Realizar novos testes comparativos com outras bases que demonstrem comportamentos diferentes (ciclo, tendência, aleatoriedade).

Referências

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

André, L. e Parpinelli, R. S. (2014). Tutorial sobre o uso de técnicas para controle de parâmetros em algoritmos de inteligência de enxame e computação evolutiva. *Revista de Informática Teórica e Aplicada*, 21(2):90–135.

https://proceedings.science/p/85353?lang=pt-br

Barreto, J. M. (2002). Introduçaoas redes neurais artificiais. *V Escola Regional de Informática. Sociedade Brasileira de Computaçao, Regional Sul, Santa Maria, Florianópolis, Maringá*, p. 5–10.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, p. 177–186. Springer.

Castañeda, W. A. C. (2016). *Metodologia de gestão ubíqua para tecnologia médico-hospitalar utilizando tecnologias pervasivas*. PhD thesis, Universidade Federal de Santa Catarina.

Ed Dlugokencky, P. T. (2017). Trends in Atmospheric Carbon Dioxide. `https://www.esrl.noaa.gov/gmd/ccgg/trends/gl_data.html`. [Online; acessado em 08/03/2018].

Gartner (2017). Gartner Says 8.4 Billion Connected "Things"Will Be in Use in 2017, Up 31 Percent From 2016. `https://www.gartner.com/newsroom/id/3598917`. [Online; acessado em 05/12/2017].

George, K. e Mutalik, P. (2017). A multiple model approach to time-series prediction using an online sequential learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–15.

Goodfellow, I., Bengio, Y., e Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., e Funaya, K. (2016). Robust online time series prediction with recurrent neural networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, p. 816–825.

Hussein, S., Chandra, R., e Sharma, A. (2016). Multi-step-ahead chaotic time series prediction using coevolutionary recurrent neural networks. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, p. 3084–3091.

Kinzler, R. (2017). Develping cognitive IoT solutions for anomaly detection by using deep learning. `https://www.ibm.com/developerworks/library/iot-deep-learning-anomaly-detection-1`. [Online; acessado em 05/12/2017].

Morettin, P. A. e de Castro Toloi, C. M. (1981). *Modelos para previsão de séries temporais*, volume 1. Instituto de Matemática Pura e Aplicada, Rio de Janeiro.

Nagelkerke, N. J. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692.

Networks, U. P. (2015). SmartMeter Energy Consumption Data in London Households. `https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households`. [Online; acessado em 08/03/2018].

Olah, C. (2014). Understanding LSTM Networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. [Online; acessado em 06/12/2017].

Queensland Government (2015). Ambient estuarine water quality monitoring data (includes near real-time sites) - 2012 to present day. `https://data.qld.gov.au/dataset/ambient-estuarine-water-quality-monitoring-data-near-real-time-sites-2012-to-present-day`. [Online; acessado em 08/03/2018].

https://proceedings.science/p/85353?lang=pt-br

Smola, A. J. e Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

Sundermeyer, M., Ney, H., e Schlüter, R. (2015). From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529. ISSN 2329-9290.

Vito, S. D., Massera, E., Piga, M., Martinotto, L., e Francia, G. D. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757.

Willmott, C. J. e Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1): 79–82.

Wu, J. e Wei, S. (1989). *Time series analysis*. Hunan Science and Technology Press, ChangSha.

Xu, M., Han, M., e Wang, X. (2016). Hierarchical neural networks for multivariate time series prediction. In *2016 35th Chinese Control Conference (CCC)*, p. 6971–6976.

https://proceedings.science/p/85353?lang=pt-br