

**PATRICK MAGALHÃES DE LIMA**

**REDES NEURAIS PARA PREDIÇÃO DE SÉRIES TEMPORAIS DE  
PRECIPITAÇÃO EM MANAUS, AMAZONAS**

Trabalho de Conclusão de Curso apresentado  
à banca avaliadora do Curso de Engenharia  
de Computação, da Escola Superior de  
Tecnologia, da Universidade do Estado do  
Amazonas, como pré-requisito para obtenção  
do título de Engenheiro de Computação.

Orientador(a): Profa. Dra. Elloá Barreto Guedes da Costa

Co-Orientador(a): Profa. Dra. Maria Betânia Leal de Oliveira

Manaus – Dezembro – 2016

***Universidade do Estado do Amazonas - UEA***  
***Escola Superior de Tecnologia - EST***

*Reitor:*

***Cleinaldo de Almeida Costa***

*Vice-Reitor:*

***Mário Augusto Bessa de Figueirêdo***

*Diretor da Escola Superior de Tecnologia:*

***Roberto Higino Pereira da Silva***

*Coordenador do Curso de Engenharia de Computação:*

***Raimundo Corrêa de Oliveira***

*Coordenador da Disciplina Projeto Final:*

***Raimundo Corrêa de Oliveira***

*Banca Avaliadora composta por:*

*Data da Defesa: 07/12/2016.*

***Profa. Dra. Elloá Barreto Guedes da Costa (Orientadora)***

***Prof. D.Sc. Carlos Maurício Figueiredo***

***Prof. M.Sc. Prof. Marcos Filipe Alves Salame***

**CIP – Catalogação na Publicação**

L864a LIMA, Patrick Magalhães de

Redes Neurais para Predição de Séries Temporais de Precipitação em Manaus, Amazonas / Patrick Magalhães de Lima; [orientado por] Profa. Dra. Elloá Barreto Guedes da Costa; [co-orientado por] Profa. Dra. Maria Betânia Leal de Oliveira – Manaus: UEA, 2016.

92 p.: il.; 30cm

Inclui Bibliografia

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação).

Universidade do Estado do Amazonas, 2016.

CDU: \_\_\_\_\_

**PATRICK MAGALHÃES DE LIMA**

**REDES NEURAIS PARA PREDIÇÃO DE SÉRIES TEMPORAIS DE  
PRECIPITAÇÃO EM MANAUS, AMAZONAS**

Trabalho de Conclusão de Curso apresentado  
à banca avaliadora do Curso de Engenharia  
de Computação, da Escola Superior de  
Tecnologia, da Universidade do Estado do  
Amazonas, como pré-requisito para obtenção  
do título de Engenheiro de Computação.

**Aprovado em: 05/12/2016**

BANCA EXAMINADORA

---

**Profa. Dra. Elloá Barreto Guedes da Costa**

*UNIVERSIDADE DO ESTADO DO AMAZONAS*

---

**Prof. Carlos Maurício Figueiredo, DsC**

*UNIVERSIDADE DO ESTADO DO AMAZONAS*

---

**Prof. Marcos Filipe Alves Salame, MsC**

*Empresa Brasileira de Pesquisa Agropecuária (Embrapa)*

# Resumo

Este trabalho aborda a predição da ocorrência de precipitação de chuva na cidade de Manaus, Amazonas. Para tanto, a abordagem de Inteligência Computacional foi considerada. Nesta abordagem, dados meteorológicos de 40 anos advindos de um estação meteorológica localizada na cidade de Manaus foram utilizados para treinar redes neurais, as quais foram adotadas por serem dirigidas a dados, capazes de detectar relações complexas e não-lineares entre variáveis e por serem capazes de generalizar. Em acordânciam com as práticas mais adotadas na literatura, redes neurais com entradas defasadas (*Time-delay Neural Network*) foram utilizadas com o objetivo de incorporar as características temporais inerentes aos dados. Para tanto, foi elaborada uma metodologia de dimensionamento das redes neurais, a qual resultou em 38 diferentes redes neurais aptas a realizar a tarefa de predição proposta. Após os treinos e testes destas redes, por meio de critérios de performance, foi possível identificar redes neurais para o problema considerado que possuem acurácia superior a 99.7%. Estes resultados mostram-se satisfatórios, provendo resultados certos a respeito da ocorrência de precipitação com uma alta porcentagem de respostas corretas.

Palavras Chave: Redes Neurais Artificiais, Séries Temporais, Inteligência Computacional.

# Abstract

*This work address the rainfall occurrence forecasting in Manaus, Amazonas. To do so, the Computational Intelligence approach was taken under consideration. According to it, meteorological data of 40 years from an automated weather station were adopted to train neural networks. This model was considered for being data-driven, able to capture complex and non-linear relationships among variables and for being capable to generalize. In accordance with the literature, time-delay neural network were preferred aiming at capturing temporal characteristics of data. A neural network dimensioning methodology was designed and applyed, producing 38 different networks to the prediction task considered. After training and testing, according to performance criteria, it was possible to identify that the most adequate neural networks for the scenario considered have accuracy higher than 99.7%. The results obtained can be considered satisfactory, delivering correct predictions of rainfall occurrence with high precision.*

Keywords: Neural networks, Time series, Computational Intelligence.

# Sumário

<b>Lista de Tabelas</b>	<b>vii</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Metodologia . . . . .	3
1.4 Organização da Monografia . . . . .	4
<b>2 Fundamentação Teórica</b>	<b>5</b>
2.1 Inteligência Computacional . . . . .	5
2.2 Redes Neurais Artificiais . . . . .	6
2.2.1 Neurônio Artificial . . . . .	7
2.2.2 Topologias de Redes Neurais Artificiais . . . . .	9
2.2.3 Aprendizagem nas Redes Neurais Artificiais . . . . .	11
2.2.4 Métodos de Avaliação de RNAs . . . . .	16
2.2.5 Redes Neurais e Séries Temporais . . . . .	18
2.3 Tecnologias Utilizadas . . . . .	21
2.4 Trabalhos Relacionados . . . . .	22
<b>3 Previsão da Precipitação em Manaus, Amazonas</b>	<b>27</b>
3.1 Caracterização do Cenário Considerado . . . . .	27

3.2	Visão Geral da Base de Dados . . . . .	29
3.3	Dimensionamento das Redes Neurais Artificiais . . . . .	31
3.4	Resultados . . . . .	33
3.5	Discussão . . . . .	34
<b>4</b>	<b>Considerações Finais</b>	<b>40</b>
4.1	Visão Geral dos Resultados e Conclusões . . . . .	40
4.2	Publicações Realizadas . . . . .	41
4.3	Trabalhos Futuros . . . . .	42
<b>A</b>	<b>Artigos Publicados</b>	<b>47</b>

# **Lista de Tabelas**

3.1	Média e desvio padrão das variáveis consideradas. . . . .	29
3.2	Coeficiente de correlação de Pearson dos atributos de entrada <i>versus</i> atributo de interesse. . . . .	30
3.3	Arquiteturas das redes neurais obtidas. . . . .	34
3.4	Resultados para os valores de MSE, MAPE e AIC de cada arquitetura de rede neural obtida. O termo “Arqt.” denota a arquitetura das redes neurais. . . . .	34

# **Lista de Figuras**

2.1	Representação de um neurônio artificial.	8
2.2	Representação de uma MLP básica, com 3 camadas.	10
2.3	Modelo abstrato de uma rede neural para efetuar predição de séries temporais.	20
3.1	Estação meteorológica automática utilizada pelo INMET.	28
3.2	Boxplots obtidos a partir dos valores do conjunto de dados.	30
3.3	Gráfico de precipitação dos últimos 60 dias da base de dados. Fonte: Elaborada pelo autor.	31
3.4	Histograma e boxplot para os valores resultantes do MSE.	35
3.5	Tamanho da janela versus valor do MSE.	36
3.6	Histograma e boxplot para os valores resultantes do MAPE.	37
3.7	Histograma e boxplot para os valores resultantes do AIC.	38
3.8	Resultados das melhores redes neurais observadas e um comparativo entre eles.	39

# Capítulo 1

## Introdução

Ao longo deste capítulo serão mostrados os elementos que apresentam este trabalho. A contextualização e a apresentação do problema são apresentados na Seção 1.1. Os objetivos gerais e específicos que guiaram a realização da pesquisa e a metodologia adotada são apresentados nas Seções 1.2 e 1.3, respectivamente. Por fim, a estrutura do restante deste trabalho é apresentada na Seção 1.4.

### 1.1 Contextualização

Manaus é a capital do Estado do Amazonas no Norte do Brasil e está localizada na região do baixo platô da planície do Rio Negro. Fundada por volta de 1693, hoje é a 7<sup>a</sup> cidade mais populosa do país com cerca de 2 milhões de habitantes no ano de 2014 (MOREIRA; SENE, 2010).

A cidade de Manaus e seus arredores possuem condições climáticas naturalmente desagradáveis, com calor e umidade típicos de regiões equatoriais. Estas características são ocasionadas, principalmente, pela localização geográfica e topografia (SIOLI, 1991). Neste cenário, tem-se a ocorrência de precipitações como um dos principais fatores para a regulação do clima local e geração de conforto para seus habitantes (SILVA, 2012).

No âmbito da Meteorologia, entende-se por precipitação o fenômeno que consiste na água proveniente da condensação atmosférica que é depositada na superfície terrestre, ocorrendo sob

a forma de chuva, granizo, neblina, neve, orvalho ou geada (TORRES; MACHADO, 2012). Desta forma, no contexto da cidade de Manaus, especifica-se a precipitação de chuva como o grande fator influente no clima da região. Ao longo deste trabalho, para fins de simplificação, será utilizado o termo “precipitação” para denotar a precipitação de chuva na cidade de Manaus.

Além do conforto para os habitantes, a precipitação em Manaus tem um papel muito mais amplo. No âmbito da Agricultura, por exemplo, muitas culturas de subsistência praticadas nesta região dependem da ocorrência de precipitações para produção. Um bom regime de chuvas pode vir a colaborar positivamente nos aspectos alimentar e econômico de diversas pessoas (ADAMS; MURRIETA; SANCHES, 2005). Outros papéis relevantes da precipitação também podem ser identificados na criação de animais, no controle e prevenção de enchentes em zonas urbanas e também no planejamento da utilização de recursos hídricos.

Dada a importância da precipitação, a tarefa de prever a sua ocorrência torna-se de grande interesse por diversos motivos. Esta predição é geralmente feita empregando-se modelos estatísticos, os quais são elaborados levando em consideração as características climáticas particulares de cada localidade. Apesar da sensibilidade à ruídos presentes nos dados, bem como dificuldade na identificação de padrões não-lineares, estes modelos produzem, em geral, bons resultados (PALIT; POPOVIC, 2005). Ainda assim, o emprego de novas técnicas e a construção de novos modelos está em constante investigação.

A Inteligência Computacional tem sido usada para endereçar a previsão de precipitação em muitos lugares do mundo. As redes neurais, em particular, tem atraído a atenção como boas preditoras por serem dirigidas a dados, capazes de generalizar, computacionalmente eficientes e poderem detectar relações não-lineares entre variáveis de entrada e saída (DARJI; DABHI; PRAJAPATI, 2015). A crescente aplicação das redes neurais na previsão de dados meteorológicos e os resultados satisfatórios obtidos em diversos lugares do mundo chamam a atenção para uma lacuna identificada:

*Aplicações da Inteligência Computacional para previsão de precipitações em Manaus  
não são conhecidas.*

Colaborar na minimização desta lacuna pode trazer diversas contribuições para a cidade e

seus habitantes, haja visto os cenários diretamente influenciados pela chuva. Portanto, considerando o papel da Inteligência Computacional para a predição de precipitação, neste trabalho serão empregadas redes neurais artificiais e séries temporais para examinar o problema da previsão da ocorrência de precipitações em Manaus.

## 1.2 Objetivos

O objetivo geral deste trabalho foi realizar a previsão da ocorrência de precipitações em Manaus utilizando redes neurais artificiais. Para alcançar esta meta, fez-se necessário contemplar alguns objetivos específicos, a citar:

1. Formular um referencial teórico que caracterize as redes neurais artificiais, em especial aquelas voltadas para previsão de séries temporais;
2. Elaborar uma base de dados meteorológicos para a cidade de Manaus a partir de dados previamente coletados por órgãos competentes;
3. Propor, treinar e testar diferentes redes neurais para prever a ocorrência de precipitações em Manaus;
4. Avaliar as redes neurais propostas para o problema em questão.

## 1.3 Metodologia

Para atingir os objetivos propostos no escopo deste trabalho, a condução das atividades seguiu a metodologia apresentada nesta seção, composta dos seguintes passos:

1. Consolidação de um referencial teórico sobre redes neurais artificiais e aplicações deste modelo na previsão de séries temporais;
2. Construção de uma base de dados sobre precipitação em Manaus a partir de bancos de dados meteorológicos;

3. Análise estatística das variáveis presentes na base de dados, a fim de descrever a relação destas com o fenômeno que se deseja prever;
4. Divisão da base de dados em conjunto de treinamento, validação e testes;
5. Dimensionamento de redes neurais para o problema em questão;
6. Treino, validação e teste das redes neurais para a previsão de precipitações em Manaus;
7. Análise de performance das redes neurais consideradas no domínio do problema investigado.

## **1.4 Organização da Monografia**

Para apresentar os resultados deste trabalho de conclusão de curso, esta monografia está organizada como segue. O Capítulo 2 apresenta conceitos, métodos e técnicas da Inteligência Computacional utilizados neste trabalho. O Capítulo 3 discorre o estudo realizado na cidade de Manaus, apresentando os detalhes de sua realização, bem como seus resultados. Por fim, no Capítulo 4 encontram-se algumas considerações a respeito deste trabalho.

# **Capítulo 2**

## **Fundamentação Teórica**

Neste capítulo serão apresentados alguns conceitos pertinentes à Inteligência Computacional, bem como métodos e técnicas comumente empregados nesta área, a fim de proporcionar melhor entendimento a respeito do trabalho realizado. Os conceitos da Inteligência Computacional e suas origens são apresentados na Seção 2.1. As redes neurais artificiais, seus principais conceitos, arquiteturas, métodos de aprendizagem e avaliação além de sua utilização com séries temporais são discorridos na Seção 2.2. As tecnologias utilizadas para a realização deste trabalho são descritas na Seção 2.3. Alguns trabalhos encontrados na literatura a respeito do problema da predição de precipitação são explanados na Seção 2.4.

### **2.1 Inteligência Computacional**

Entende-se por Inteligência Computacional (IC) um conjunto de metodologias computacionais inteligentes aplicadas com o intuito de resolver problemas complexos cuja solução é não-trivial, ou mesmo intratável, utilizando-se os métodos computacionais convencionais (PALIT; POPOVIC, 2005). Muito foi discutido quanto a uma definição exata da IC, desde que foi inicialmente caracterizada por lidar principalmente com o processamento de dados numéricos em contraste com a Inteligência Artificial que em grande parte lida com conhecimento não numérico (BEZ-DEK, 1992). Ainda assim, a delimitação das disciplinas associadas à IC não está totalmente clara.

A origem da IC envolve-se com o que é conhecido por *Soft Computing*, definido por seu criador como uma coleção de metodologias que permitem explorar a tolerância à imprecisão e incerteza, objetivando tratabilidade, robustez e baixo custo computacional (ZADEH, 1994). Os constituintes principais desta área do conhecimento compreendem a Lógica *Fuzzy*, Neuro-computação, Raciocínio Probabilístico, Computação Evolucionária, Teoria da Confiabilidade e Teoria do Aprendizado.

Apesar destas duas áreas, Inteligência Computacional e *Soft Computing*, terem evoluído praticamente em paralelo desde suas definições, esta ultima é considerada como o cerne da Inteligência Computacional e incorpora métodos probabilísticos. Além disso, o uso de outros paradigmas, como *Swarm Intelligence* e Sistemas Imunológicos Artificiais, por parte da IC ajuda a diferenciar os dois campos (ENGELBRECHT, 2007).

Técnicas da IC têm sido eficientemente utilizadas em aplicações de análise e predição de séries temporais, mineração de dados, reconhecimento de padrões, sistemas de controle, sistemas de diagnóstico de falhas, otimização, dentre outros problemas tanto no âmbito industrial quanto acadêmico (PALIT; POPOVIC, 2005).

## 2.2 Redes Neurais Artificiais

Dentre as ferramentas da IC, as *Redes Neurais Artificiais* (RNAs) originam-se do que é conhecido como abordagem conexionista da Inteligência Artificial sendo formalmente definidas como sistemas paralelamente distribuídos compostos por unidades de processamento relativamente simples que computam funções matemáticas para produzirem resultados (HAYKIN, 2009; ENGELBRECHT, 2007; BRAGA; CARVALHO; LUDEMIR, 2007).

As RNAs surgiram como uma tentativa de modelar a capacidade de processamento de informação do sistema nervoso humano e seu principal constituinte, o cérebro, cujas unidades básicas, os neurônios, são dispostas de maneira interconectada formando uma gigantesca rede orgânica, que opera com uma essência de “controle por meio da comunicação” (ROJAS, 1998), em que cada célula lida com vários estímulos de entrada de formas diversas, produzindo sinais de saída responsáveis por realizar tarefas complexas, tais como o reconhecimento de padrões.

As unidades básicas de processamento de uma RNA são os *neurônios artificiais*, modelos matemáticos que buscam abstrair o funcionamento de seus equivalentes orgânicos, os neurônios presentes no cérebro. Assim como no cérebro, os neurônios artificiais são dispostos de forma interconectada e em camadas, definindo o que conhece-se como *topologia* de uma RNA. Outra característica é que geralmente os neurônios artificiais possuem pesos associados às suas conexões de entrada, que podem ser interpretados como a importância ou influência desta, no que diz respeito à computação do neurônio. O funcionamento do neurônio artificial e a organização das RNAs em camadas são detalhados em subseções posteriores.

Atualmente redes neurais artificiais são amplamente usadas nas mais diversas aplicações pelas seguintes razões: (i) RNAs são dirigidas por dados (*data driven*) e não demandam pré-requisitos restringentes sobre o que está sendo modelado; (ii) RNAs podem predizer padrões que não são fornecidos durante o treinamento, isto é, são capazes de generalizar; (iii) RNAs são eficientes no treinamento de grandes amostras de dados graças a sua capacidade de processar em paralelo; (iv) RNAs possuem a habilidade de detectar relações complexas e não-lineares entre as variáveis dependentes e independentes (DARJI; DABHI; PRAJAPATI, 2015).

Vale também ressaltar sua capacidade de aproximar qualquer função contínua ou não-contínua (CYBENKO, 1989), o que lhes concedeu o título de aproximador universal, além de sua notória tolerância a falhas e possibilidade de implementação usando a tecnologia *very-large-scale-integrated* (HAYKIN, 2009).

### 2.2.1 Neurônio Artificial

O *neurônio artificial* é a unidade fundamental na construção de redes neurais artificiais. O primeiro modelo, conhecido por neurônio MCP devido a seus criadores McCulloch e Pitts, apesar de possuir uma representação simples em virtude de suas entradas e saídas binárias, tinha o objetivo de imitar a realidade biológica (MCCULLOCH; PITTS, 1943). Avanços significativos adviriam com os trabalhos sobre aprendizagem de Donald Hebb e principalmente com a publicação do modelo *perceptron* (ROSENBLATT, 1958), que resolia problemas linearmente separáveis e introduziu o conceito de aprendizado em RNAs.

Uma generalização do modelo de um neurônio artificial é apresentado na Figura 2.1. Neste modelo, existe um conjunto de  $m$  terminais de entrada, representando os dendritos de um neurônio biológico, por onde sinais são introduzidos. Associa-se um peso a cada entrada, representando a relevância referente a uma conexão sináptica. Há também o peso  $w_0$ , um termo de polarização criado com a intenção de estabelecer um limiar de ativação para cada neurônio. Este peso correspondente a entrada *bias*, cujo valor é sempre unitário. Pode-se então definir um vetor de entradas  $X = [+1, x_1, x_2, \dots, x_m]$  e um vetor de pesos  $W = [w_0, w_1, w_2, \dots, w_m]$ . As entradas e pesos são combinados por meio de uma função  $\Phi : \mathbb{R}^{m+1} \rightarrow \mathbb{R}$ , que geralmente é a soma ponderada das entradas e dos pesos, como na Equação 2.1

$$\Phi(X, W) = \sum_{i=0}^m x_i \cdot w_i \quad (2.1)$$

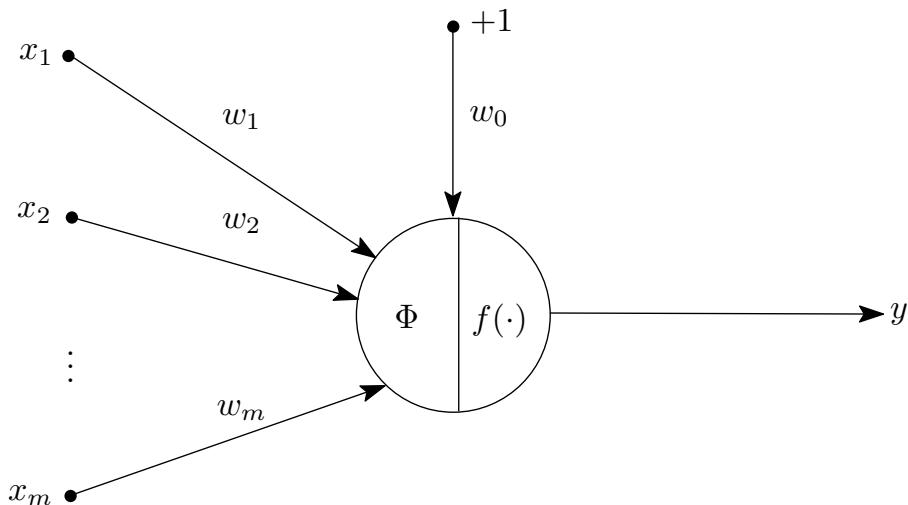


Figura 2.1: Representação de um neurônio artificial.

A função  $f$  é chamada função de ativação e fornece a resposta de um neurônio para uma dada entrada. É monotônica e contínua, comumente possuindo forma sigmoidal (MANDIC; CHAMBERS, 2001). É com ela que a computação de um neurônio pode adquirir características não-lineares, sendo as funções sigmoidal e tangente hiperbólica mais comumente usadas. Portanto, o valor da saída  $y$  do neurônio, será obtido por meio da Equação 2.2.

$$y = f(\Phi(X, W)) \quad (2.2)$$

Utilizando  $u = \sum_{i=1}^m (x_i \cdot w_i) + w_0$ , o que torna a saída de um neurônio igual a  $y = f(u)$ , é possível notar a influência do termo de polarização  $w_0$  no aumento ou diminuição do valor líquido da entrada da função de ativação, de forma a transladar o valor de  $f$  no eixo  $u$ .

Neurônios artificiais tem uma capacidade computacional limitada, independente da função de ativação escolhida. No entanto, um conjunto de neurônios artificiais conectados na forma de uma rede – uma rede neural – adquirem a capacidade de resolver problemas de elevada complexidade (BRAGA; CARVALHO; LUDELMIR, 2007).

### 2.2.2 Topologias de Redes Neurais Artificiais

As diferentes formas de conexão entre os neurônios artificiais definem a *topologia* de uma rede neural artificial. RNAs também podem ser especificados pela característica de seus neurônios e regras de treinamento ou aprendizado. Dois grandes grupos podem ser definidos quando se analisam as RNAs por seus padrões de conexões, sendo eles: (i) redes *feedforward*, nas quais não existem ciclos; (ii) redes recorrentes nas quais podem haver ciclos, ou seja, a saída de pelo menos um neurônio alimenta a entrada de algum outro neurônio (MANDIC; CHAMBERS, 2001).

As redes *feedforward* podem ser interpretadas como um grafo direcionado não-cíclico, que propaga os sinais recebidos em uma camada para a camada posterior. Tal propagação inicia-se da camada de entrada e sucede-se até que a camada de saída forneça uma resposta. As redes Perceptron de Múltiplas camadas (MLP, do inglês *Multilayer Perceptron*) são as redes *feedforward* mais difundidas e empregadas. Estas redes são compostas por múltiplas camadas cujos neurônios possuem função de ativação sigmoidal em suas camadas internas, sendo capazes de efetuar o mapeamento entrada-saída de qualquer função arbitrária, qualificando-as como eficientes para diversas tarefas de previsão e classificação (PALIT; POPOVIC, 2005).

A representação de uma MLP é mostrada na Figura 2.2. Nesta figura pode-se observar a estrutura básica destas redes, compostas por (i) uma camada de entrada, com  $m$  unidades,

por onde as informações são apresentadas à rede; (ii) uma camada de saída, que fornece  $n$  valores como resultado da computação da rede; (iii) uma camada escondida (*hidden layer*), ou intermediária, com  $k$  neurônios responsáveis por extrair características inerentes aos dados computados pela rede, realizando o mapeamento entrada-saída propriamente dito. Os neurônios estão identificados por  $a_j^\ell$ , em que  $\ell$  refere-se a camada de onde o neurônio pertence e  $j$  é o índice do neurônio. Também estão representadas as entradas *bias*, uma em cada camada da rede com exceção da camada de saída, uma vez que os resultados produzidos por esta não passarão por outros neurônios. Um detalhe importante é que as unidades da primeira camada da rede não realizam computações, somente repassando os valores numéricos recebidos para as camadas subsequentes.

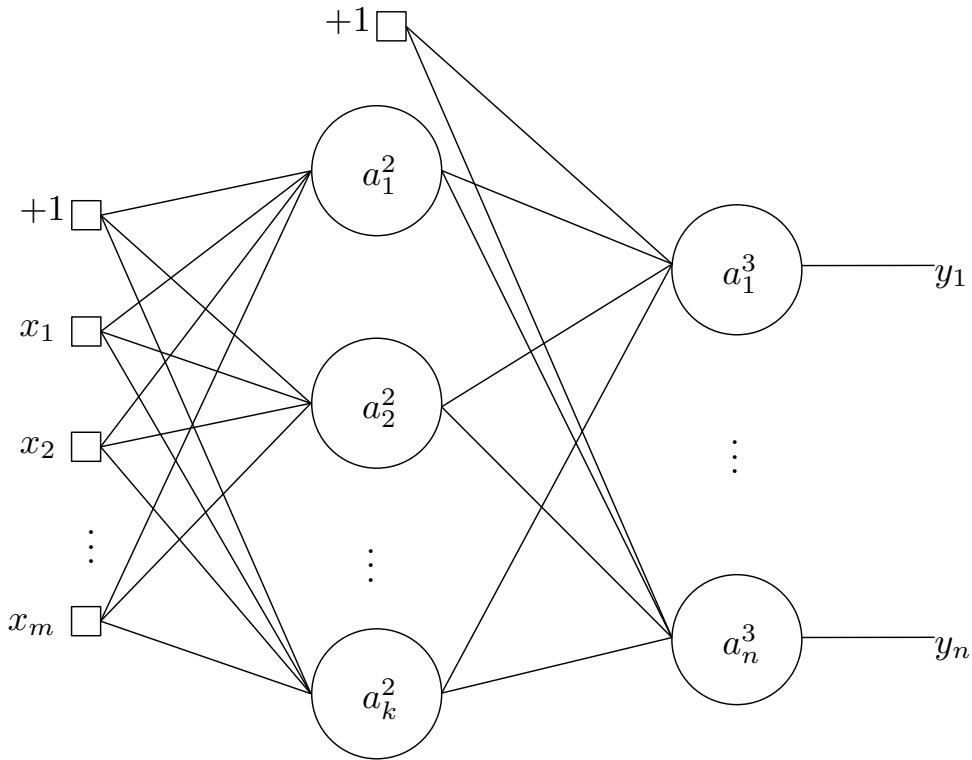


Figura 2.2: Representação de uma MLP básica, com 3 camadas.

Embora não haja restrição para a quantidade de camadas intermediárias de uma rede neural, a maioria dos problemas práticos raramente necessita de mais de uma camada intermediária, sendo esta necessária somente quando a função a ser aproximada apresenta descontinuidades. Uma vez que as camadas intermediárias trabalham com estimativas de erro, quando presentes em número superior ao necessário, estas podem dificultar a convergência global da rede

(BRAGA; CARVALHO; LUDERMIR, 2007).

### 2.2.3 Aprendizagem nas Redes Neurais Artificiais

Uma importante característica das RNAs é a sua capacidade de aprender por meio de exemplos. O processo de *aprendizado* de uma rede neural, no qual são extraídas características relevantes de dados de um determinado domínio, consiste em sucessivos ajustes dos pesos associados a seus neurônios, de modo a aprimorar seu desempenho de acordo com algum critério pré-estabelecido. Tais ajustes são efetuados por meio de um conjunto de regras bem definidas que especificam quando e como o valor de cada peso deve ser alterado, enquanto a rede processa um *conjunto de dados* que representa uma amostra do problema a ser resolvido. Este conjunto de regras que regem a alteração dos pesos de uma rede neural é formalmente conhecido como *algoritmo de aprendizado* ou *regra de aprendizado* (FACELI et al., 2011), que pode ser expresso de maneira genérica por:

$$w(t + 1) = w(t) + \Delta w(t), \quad (2.3)$$

em que  $w(t)$  representa o valor de um peso no instante de tempo  $t$  (antes de ser ajustado),  $\Delta w(t)$  representa seu *valor de ajuste*, enquanto  $w(t + 1)$  denota o valor do peso após ser ajustado, no instante  $t + 1$ .

Diversos algoritmos de aprendizado foram propostos, diferindo basicamente na maneira como  $\Delta w(t)$  é calculado, podendo estes ser agrupados em dois *paradigmas de aprendizado*, que dizem respeito à maneira como o processo de aprendizagem de uma rede neural é conduzido. Estes paradigmas são apresentados a seguir.

#### Aprendizado Supervisionado

Neste paradigma, a rede aprende por meio de pares entrada-saída, ou seja, por meio de um conjunto de dados a rede adquire conhecimento para mapear uma determinada entrada para sua saída respectiva. O nome deste paradigma advém da presença de um supervisor que avalia a resposta da rede. Este aprendizado é comumente empregado em problemas de aproximação ou predição e pode ser implementado de duas maneiras: (i) *on-line* em que os pesos são ajustados

a cada exemplo apresentado à rede; (ii) *off-line* ou *batch* na qual os ajustes de pesos ocorrem após a apresentação de todos os exemplos à rede, o que caracteriza uma época (*epoch*) de treinamento (HAYKIN, 2009). Os principais tipos de algoritmo de treinamento característicos deste paradigma são:

- **Correção de erros.** Ajustam os pesos de uma RNA de forma a reduzir o erro da resposta da rede com relação à saída desejada. O ajuste de pesos destes algoritmos deve ser proporcional ao produto do valor de entrada de um neurônio pelo erro associado a ele, podendo ser denotado genericamente por:

$$w(t+1) = w(t) + \eta \cdot e(t) \cdot x(t), \quad (2.4)$$

em que  $w(t)$  é o peso associado ao valor de entrada  $x(t)$ ,  $e(t)$  é uma medida de erro da rede e  $\eta$  é um valor de proporcionalidade conhecido por taxa de aprendizado;

- **Aprendizado por reforço.** Considerado um caso particular de aprendizado supervisionado, consiste em “recompensar” a rede por bons resultados e “puni-la” ao produzir resultados ruins, caracterizando-se como um processo de tentativa e erro cujo objetivo é maximizar um índice de desempenho escalar chamado sinal de reforço (BRAGA; CARVALHO; LUDELMIR, 2007).

## Aprendizado Não-Supervisionado

As redes cujo aprendizado é regido por este paradigma são fornecidos apenas dados de entrada, com o objetivo de que esta extraia características estatisticamente relevantes destes dados, ou seja, não é avaliado se a resposta da rede é correta ou não, mas estimula-se a produção de respostas semelhantes, por parte da rede, para entradas semelhantes, sendo por isso geralmente empregado em problemas de classificação e categorização (BARRETO, 2002). No contexto deste paradigma os principais tipos de algoritmos de aprendizado são:

- **Aprendizado Hebbiano.** Também conhecido como regra de Hebb, foi definido por Donald Hebb (HEBB, 1949) e determina que se dois neurônios estão simultaneamente

ativados, então a conexão entre eles deve ser fortalecida caso contrário, deve-se enfraquecer-la. A atualização dos pesos é obtida conforme a Equação 2.3, sendo seu valor de ajuste definido por:

$$\Delta w(t) = \eta x(t)o(t), \quad (2.5)$$

em que  $\eta$  é a taxa de aprendizado,  $x(t)$  é o valor de entrada apresentado e  $o(t)$  é a resposta da rede a esta entrada, uma vez que não há um valor de saída desejado. Apesar de mais comumente usado em aprendizado não-supervisionado, é possível utilizar o aprendizado Hebbiano com o paradigma supervisionado, no qual  $o(t)$  assume o valor da saída desejada para uma determinada entrada (RUTKOWSKI, 2008);

- **Aprendizado por competição.** Estes algoritmos promovem uma competição entre os neurônios de saída por suas respectivas ativações dado um determinado sinal de entrada. A unidade de saída vencedora tem então seus pesos ajustados. Esta pode ser interpretada como a mais apta para sinalizar um padrão de entrada (OIKAWA; ISHIKI, 2013).

Nota-se, portanto, que o processo definido como *treinamento* de uma rede neural envolve a escolha dos seguintes elementos: (i) um paradigma de aprendizado, que varia de acordo com o problema considerado; (ii) um algoritmo de aprendizado que esteja em acordo com o item anterior; (iii) uma métrica de desempenho para avaliar o aprendizado da rede neural, consistindo geralmente no erro associado à sua resposta.

É importante salientar que no aprendizado supervisionado de redes com múltiplas camadas os únicos neurônios cujas saídas podem ser acessadas diretamente estão presentes na última camada da rede, possibilitando a obtenção do erro associado às respostas destes neurônios e, consequentemente, o ajuste de seus pesos. O ajuste dos pesos dos neurônios presentes nas camadas intermediárias de uma RNA é igualmente necessário. A obtenção do erro associado a estes neurônios, porém, caracteriza um problema para o aprendizado, pois este valor não pode ser inferido diretamente de suas saídas, uma vez que estas representam uma resposta parcial da rede. Este problema é sanado com a utilização do algoritmo de retropropagação de erros (*backpropagation*), amplamente utilizado para o treinamento supervisionado de RNAs.

## Backpropagation

O algoritmo de aprendizado *backpropagation* é do tipo correção de erros sendo amplamente empregado no aprendizado supervisionado de RNAs de múltiplas camadas. Seu funcionamento consiste em utilizar o método do *gradiente descendente* para ajustar os pesos de uma RNA de modo a minimizar uma *função de erro*. Tal função de erro atua no espaço do vetor de pesos sinápticos da RNA (ROJAS, 1998). O treinamento de redes neurais utilizando o algoritmo *backpropagation* caracteriza-se pelas seguintes fases:

1. **Inicialização dos pesos.** Consiste em definir o valor inicial das conexões sinápticas entre as unidades da RNA, de modo que possuam uma distribuição uniforme. Estes valores geralmente são obtidos pela geração de números pseudo aleatórios definidos em um dado intervalo, geralmente entre 0 e 1;
2. **Apresentação dos exemplos de treino.** Aqui os dados do conjunto de treinamento são apresentados à rede neural. A literatura, indica que tal apresentação seja feita de maneira randômica. Para cada dado apresentado, realizar-se-ão as etapas a seguir:
  - 2.1. **Forward.** Nesta etapa, os pesos sinápticos permanecem inalterados enquanto uma determinada entrada é computada pela rede, sendo os resultados produzidos por cada neurônio sucessivamente propagados, até que alcancem a camada de saída e esta produza a resposta da rede àquela entrada;
  - 2.2. **Backward.** Nesta fase, um sinal de erro é produzido para cada neurônio da camada de saída da rede por meio da comparação de sua resposta à resposta desejada. Este sinal erro é então retropropagado para neurônios das camadas precedentes, através das conexões sinápticas que os ligam, de modo que estes recebam somente o valor de erro referente à sua influência na computação. Após esta retropropagação percorrer todas as camadas da rede, os erros de cada neurônio são usados, então, para o ajuste dos pesos da rede neural utilizando o método do gradiente descendente (HAYKIN, 2009).

Desta forma, ao fim do treinamento, a rede neural terá armazenado em seus pesos o conhecimento referente aos dados apresentados, de modo a produzir uma saída consistente com um valor de erro relativamente baixo no melhor caso.

A descoberta e formalização do algoritmo *backpropagation* permitiu o treinamento de redes neurais com múltiplas camadas, as quais, consequentemente, não estavam limitadas a resolver problemas linearmente separáveis, aumentando assim a utilidade destes modelos (WILAMOWSKI, 2011). Entretanto, o emprego do algoritmo *backpropagation*, embora represente um marco no treinamento de RNAs, não garante total eficiência em seu aprendizado.

Alguns fatores afetam diretamente a convergência do aprendizado das RNAs. Um deles é a inicialização dos pesos das conexões sinápticas, que pode conduzir o gradiente descendente a um mínimo local da função de erro, o que não é desejável, pois acarreta em uma performance ruim por parte da rede neural. Outro fator é a escolha de uma taxa de aprendizagem,  $\eta$  na Equação 2.3, que influí diretamente na atualização dos pesos, podendo assim alterar a convergência do treinamento显著mente. Vale ressaltar também a influência da quantidade de dados utilizados para treinar a rede neural e a qualidade dos parâmetros empregados como suas entradas, pois faz-se necessário que estes sejam suficientemente significativos com relação ao domínio do problema analisado, uma vez que todo conhecimento obtido pela rede será daí extraído (RUTKOWSKI, 2008).

Para melhorar o processo de treinamento, algumas técnicas foram empregadas para acelerar a convergência do mesmo, como a seleção do “melhor” conjunto inicial de pesos, o emprego de técnicas adequadas de pré-processamento dos dados a serem utilizados (normalização, algoritmos de extração de características, etc.), bem como taxas de aprendizados variáveis e o emprego de um termo *momentum* que adiciona um valor extra na atualização dos pesos durante o treinamento, de modo a estabilizar a convergência do gradiente descendente (HAYKIN, 2009).

Variações da versão clásica do algoritmo *backpropagation* foram desenvolvidas com o intuito de incorporar algumas das técnicas previamente citadas e aprimorar sua eficiência e eficácia tanto no contexto de aprendizado de RNAs quanto no contexto computacional. Dentre essas va-

riações pode-se citar: *Backpropagation* com *momentum*, *Resilient Backpropagation* e *Levenberg-Marquardt* que emprega uma taxa de aprendizado variável (DEMUTH; BEALE; HAGAN, 2009).

O algoritmo *backpropagation* de Levenberg-Marquardt demonstrou ser o algoritmo mais rápido para o treinamento de redes neurais *feedforward* de dimensões moderadas (KIPLI et al., 2012). Este algoritmo utiliza o método de Gauss-Newton, para avaliar a superfície de erro e variar a taxa de aprendizado  $\eta$ , de modo a aprimorar o ajuste de pesos, melhorando, deste modo, a convergência do gradiente descendente. Assim, a ideia básica do algoritmo é utilizar o método do gradiente descendente para minimizar os erros cometidos pela rede neural durante seu treinamento, combinado ao ajuste dinâmico da taxa de aprendizado, realizado empregando-se o método de Gauss-Newton (WILAMOWSKI, 2011)

Além da escolha dos elementos necessários ao treinamento efetivo de uma RNA, também faz-se necessária uma maneira de mensurar a performance das redes treinadas, permitindo sua análise e comparação no intuito de selecionar o modelo de rede neural mais apropriado para determinado domínio. Esta estimativa é feita por meio de técnicas e métricas matemáticas, a serem apresentadas a seguir.

#### **2.2.4 Métodos de Avaliação de RNAs**

Ao endereçar problemas às redes neurais, além da escolha de uma topologia apropriada, é necessário dimensionar as redes, isto é, definir a quantidade de camadas escondidas e número de neurônios presentes nestas camadas, para que o treinamento de cada rede possa, então, ser realizado. O dimensionamento de RNAs não possui regra geral, sendo feito de maneira empírica. (BRAGA; CARVALHO; LUDELMIR, 2007). Deste processo resultam, geralmente, diversas redes neurais para realização de uma mesma tarefa, que possuem desempenhos diferentes. Assim, é necessária a definição de métricas que permitam avaliar a performance destas redes, com o objetivo de determinar a qualidade das respostas produzidas.

A maneira mais usual de avaliar a performance de um modelo de rede neural dá-se pela estimativa dos erros cometidos por ela ao processar uma ou várias entradas. Uma das métricas

mais utilizadas para este tipo de avaliação é o Erro Médio Quadrático (MSE, do inglês *Mean Squared Error*), expresso pela Equação 2.6. Esta métrica fornece um indicativo do quanto a saída ( $o_i$ ) de uma RNA aproxima-se de seu valor de referência ( $y_i$ ), obtendo-se o valor médio para esta proximidade, para  $n$  amostras avaliadas.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (o_i - y_i)^2 \quad (2.6)$$

O Erro Percentual Médio Absoluto (MAPE, do inglês *Mean Absolute Percentage Error*) definido pela Equação 2.7, fornece uma medida, de maneira percentual, do erro cometido pela rede. É considerada a métrica mais intuitiva, por fornecer uma estatística percentual dos erros da rede (HTIKE; KHALIFA, 2010). Além disso, a acurácia da rede neural pode ser obtida por meio do MAPE, subtraindo-se o valor MAPE de 1. A obtenção do MAPE, mostrada na Equação 2.7, considera um conjunto com  $n$  amostras, em que  $o_i$  representa a saída da rede neural e  $y_i$  seu valor de referência (real), ambos para a  $i$ -ésima amostra avaliada:

$$\text{MAPE} = \left[ \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - o_i}{y_i} \right| \right] \times 100 \quad (2.7)$$

A grande desvantagem destas métricas é que, por serem obtidas por meio do processamento de um determinado conjunto, dependem de um cenário de teste específico. Como alternativa para minimizar esta limitação, o *Critério de Informação de Akaike* (AIC, do inglês *Akaike's Information Criterion*) é uma medida frequentemente considerada.

Esta métrica fundamenta-se no conceito de entropia da Teoria da Informação, oferecendo uma medida relativa da informação perdida quando determinado modelo é usado para descrever a realidade, podendo ser considerado como a troca entre viés e variância na construção de um modelo. É uma base de comparação e seleção entre diversos modelos estatísticos, sendo aquele que possui o menor valor AIC considerado o melhor no cenário em questão (PANCHAL et al., 2010).

De maneira geral, para efetuar o cálculo para obtenção do AIC necessita-se dos seguintes parâmetros:

1. Número de neurônios de entrada ( $in$ );
2. Número de neurônios na camada oculta ( $h$ );
3. Number of neurônios de saída ( $o$ );
4. Soma residual dos quadrados (RSS, do inglês *Residual Sum of Squares*);
5. Quantidade de dados de treinamento ( $n$ ).

A partir destes parâmetros, deriva-se  $k$ :

$$k = in \cdot h + o \cdot h. \quad (2.8)$$

A partir da razão  $n/k$ , obtém-se o AIC:

$$AIC = \begin{cases} n \cdot \ln\left(\frac{RSS}{n}\right) + 2 \cdot k + \frac{2 \cdot k \cdot (k+1)}{n-k-1}, & \text{se } n/k < 40; \\ n \cdot \ln(RSS/n) + 2 \cdot k, & \text{c.c.} \end{cases} \quad (2.9)$$

O AIC pode, então, ser utilizado como uma medida da qualidade relativa de diversas redes neurais para um conjunto de dados sendo utilizado. Assim, dada uma coleção de diferentes redes neurais, o AIC atua como uma medida para estimar a qualidade de cada modelo, em relação a todos os outros.

### 2.2.5 Redes Neurais e Séries Temporais

As redes neurais podem ser aplicadas a diversos problemas. Os problemas de predição são de constante interesse, tanto no âmbito acadêmico quanto industrial, pois uma estimativa futura de certa variável pode propiciar um melhor planejamento e tomada de decisões para determinada atividade.

Neste cenário, geralmente é necessário prever os valores de interesse em determinada ordem de ocorrência, o que caracteriza uma série temporal. Uma *série temporal* é uma sequência

ordenada no tempo de observações de valores de uma variável, feita em intervalos  $\Delta t$  igualmente espaçados, representada como um conjunto de valores discretos  $x_1, x_2, x_3, \dots, x_{t-1}$ . O interesse consiste em prever o valor futuro  $x_t$  baseado nos valores anteriores da variável. Numa abordagem formal, almeja-se encontrar uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  tal que se obtenha  $x_t$  a partir de  $n$  valores anteriores ao tempo  $t$ :

$$x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-n+1}) \quad (2.10)$$

Redes neurais tem sido amplamente utilizadas como preditoras de séries temporais. Em 1964, (HU, 1964) foi o primeiro a demonstrar – em um exemplo de previsão climática – a capacidade geral de previsão das redes neurais. Desde então, muitas aplicações envolvendo previsões ligadas ao mercado financeiro, Meteorologia e tráfego de redes têm sido descritas na literatura como tendo resultados satisfatórios (FRANK; DAVEY; HUNT, 2001).

A abordagem clássica para previsão de séries temporais emprega métodos estatísticos, tais como o Box-Jenkins, que é uma metodologia de previsão geral para séries temporais gerada a partir de um processo estacionário autorregressivo de médias móveis (PALIT; POPOVIC, 2005). Contudo, abordagens estatísticas apresentam carências na identificação de padrões não-lineares e irregularidades em séries temporais (DARJI; DABHI; PRAJAPATI, 2015). Diversos trabalhos na literatura verificaram que redes neurais possuem a habilidade de, implicitamente, detectar relações não-lineares complexas entre variáveis dependentes e independentes. Como resultado, quando comparadas a métodos clássicos, pode-se verificar que: (i) para séries temporais com memória longa, ambas as abordagens produzem resultados similares; (ii) redes neurais superam a abordagem clássica Box-Jenkins em alguns experimentos por mais de 100%; e (iii) as topologias de redes neurais otimamente ajustadas possuem maior eficiência que os algoritmos tradicionais correspondentes (PALIT; POPOVIC, 2005).

Uma rede neural para previsão de séries temporais visa a induzir a função  $f$  da Equação 2.10, mas com algumas particularidades. Com o intuito de prever um certo  $x_t$ ,  $k$  parâmetros  $p_t^1, p_t^2, \dots, p_t^k - t$  coletados no instante  $t$  são fornecidos juntamente com entradas anteriores  $u_{t-1}, u_{t-2}, \dots, u_{t-w}$  e as respectivas saídas  $x_{t-1}, x_{t-2}, \dots, x_{t-w}$  do fenômeno que está sendo pre-

visto. Se o foco é na predição “um passo à frente”, então apenas um neurônio na camada de saída é necessário, o qual irá retornar  $x_t$ ; em caso contrário, na predição “múltiplos passos à frente”,  $\ell+1$  neurônios devem ser empregados na camada de saída, retornando  $x_t, x_{t+1}, \dots, x_{t+\ell}$  (BRAGA; CARVALHO; LUDERMIR, 2007). A estratégia para usar redes neurais com séries temporais é ilustrada na Figura 2.3. O número  $w$  de entradas e saídas defasadas consideradas pela rede neural caracterizará o *tamanho da janela*, termo comumente utilizado na literatura para referir-se a esta grandeza.

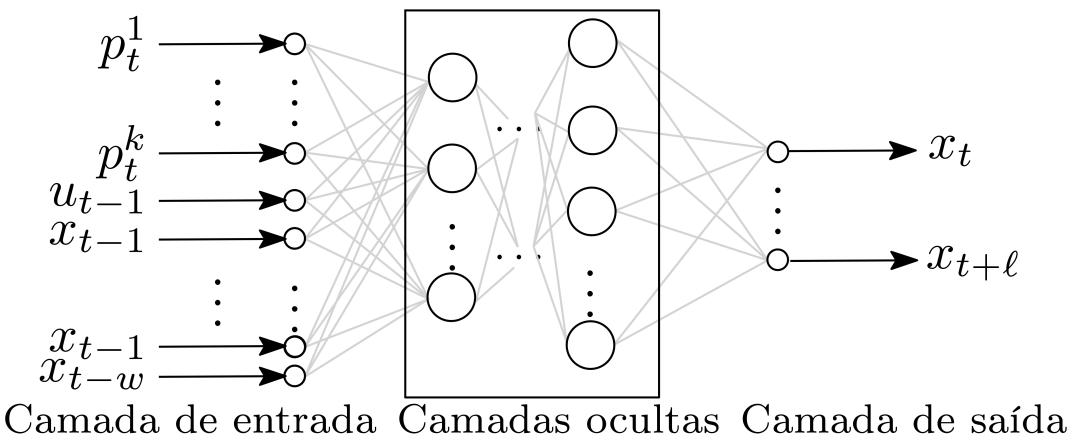


Figura 2.3: Modelo abstrato de uma rede neural para efetuar predição de séries temporais.

Determinar a arquitetura de uma rede neural no cenário de predição de séries temporais é considerado um desafio. É necessário decidir: (1) que tipo de predição será considerada (um ou múltiplos passos à frente), pois isto definirá o número de neurônios na camada de saída; (2) quantas observações, devidamente defasadas no tempo, serão empregadas como entrada (*estimativa do tamanho da janela*); (3) quais e quantos são os parâmetros fornecidos como entrada; (4) o número de camadas escondidas e de neurônios em cada camada – estes parâmetros são úteis para capturar a relação não-linear entre variáveis de entrada e saída; e (5) o tipo de rede neural (Perceptron de Múltiplas Camadas, *Radial Basis Function*, etc.).

Algumas respostas para essas questões podem ser encontradas na literatura. Redes neurais *feedforward* são majoritariamente usadas no cenário de predição de séries temporais. Com respeito ao número de camadas escondidas, a maior parte das aplicações de predição usa apenas

uma, com um pequeno número de neurônios escondidos, pois uma rede neural com esta topologia é suficientemente capaz de extrair as características presentes nos dados de entrada, de acordo com o teorema de Kolmogorov. Percebe-se que a parte mais difícil consiste em determinar os dados de entrada e a quantidade de amostras defasadas. Resultados empíricos sugerem que determinar a camada de entrada chega a ser mais importante que determinar a camada escondida em problemas de predição de séries temporais (PALIT; POPOVIC, 2005).

## 2.3 Tecnologias Utilizadas

O emprego de RNAs como paradigma para a resolução de problemas demanda, além do conhecimento dos conceitos envolvidos, uma maneira de implementá-las. A implementação de RNAs pode suceder-se essencialmente por meio de: (i) implementação por circuitos analógicos ou digitais e (ii) simulação. O primeiro caso, ao mesmo tempo que fornece um funcionamento dedicado e de fato paralelo, também agrupa uma série de complexidades, principalmente no que diz respeito à implementação dos circuitos de aprendizagem das redes neurais. O segundo caso, apesar da possibilidade de limitação de performance, incorpora facilidades como o uso de simuladores para propósitos genéricos, bem como a possibilidade de gerar diversas redes neurais com o mínimo de esforço (BARRETO, 2002). Neste trabalho, optou-se pela implementação simulada de redes neurais artificiais.

A simulação de redes neurais pode ser efetuada utilizando-se linguagens de programação de uso geral e linguagens, ou plataformas, de simulação. Por fins praticidade, optou-se por utilizar a plataforma MATLAB<sup>®</sup>, uma das mais conhecidas e utilizadas para cálculos e resolução de problemas nas Engenharias e Ciências. Esta plataforma provê um processamento otimizado, uma linguagem de programação baseada em matrizes, além de um conjunto de *toolboxes* para os mais variados propósitos. Este último recurso oferecido é de grande interesse no contexto deste trabalho, uma vez que a plataforma MATLAB disponibiliza uma ferramenta propícias à simulação de RNAs.

O *Neural Network Toolbox* é um ambiente oferecido pela plataforma MATLAB que provê algoritmos e funcionalidades que fomentam a construção, treino, visualização e simulação de

redes neurais artificiais. Este ambiente fornece suporte à diversas redes neurais as quais podem ser aplicadas a diversas problemáticas tais como classificação, regressão, clusterização, predição de séries temporais, dentre outras (MATHWORKS, 2016). São disponibilizados também alguns algoritmos de pré e pós-processamento de dados e diversos algoritmos de treinamento implementados.

## 2.4 Trabalhos Relacionados

Redes neurais têm sido adotadas para a predição de precipitação em diversos trabalhos na literatura, provendo resultados relevantes e sendo considerada uma abordagem mais apropriada que os métodos clássicos comumente usados em tarefas de predição (NAYAK; MAHAPATRA; MISHRA, 2013). Nesta seção serão apresentados alguns trabalhos da literatura que utilizam redes neurais como abordagem para a predição de precipitação. Algumas características ou resultados destas referências que venham a corroborar com o trabalho em questão serão adequadamente enfatizadas.

O trabalho (ABBOT; MAROHASY, 2012) considerou a precipitação mensal e sazonal em Queensland, Australia. Os autores desenvolveram Redes Neurais *Time-Delay Recorrentes* (RNTDR) para prever a ocorrência de precipitação 3 meses à frente. Os dados de entrada foram divididos em quatro classes: (*i*) precipitação mensal, correspondente a 20 pontos no estado de Queensland, defasados num período de 12 meses; (*ii*) índices climáticos, dos quais são relevantes o Índice Oscilação Sul (SOI), Oscilação Decadal do Pacífico (PDO), Niño 3.4 e Modo de Índice Dipolo (DMI); (*iii*) temperatura atmosférica, considerando seus valores mínimos e máximos; (*iv*) dados solares, dos quais foram escolhidos mancha solar e irradiação solar total.

Os autores desenvolveram uma RNTDR com duas camadas escondidas usando uma abordagem de tentativa e erro para definir e avaliar diferentes configurações. Os modelos resultantes foram usados para prever a ocorrência de chuva em cada um dos 20 lugares usando o Erro Médio Quadrático Normalizado (NMSE, do inglês *Normalized Mean Square Error*) e o coeficiente de correlação de Pearson como estimadores de performance. Inicialmente, somente os valores de precipitação mensal foram usados como entradas, adicionando-se os outros dados

progressivamente. A melhor combinação de dados de entrada consistiu dos valores de precipitação mensais defasados, temperatura atmosférica, SOI, PDO, e Niño 3.4. Os resultados de performance da RNTDR foram então comparados com o Modelo de Predição de Atmosfera Oceânica para a Austrália (POAMA) e com o Modelo de Circulação Geral (GCM). Os modelos RNTDR desenvolvidos obtiveram um valor de NMSE mais baixo em 17 lugares de um total de 18 usados nesta comparação, comprovando o bom desempenho das redes neurais em tarefas de predição e demonstrando sua boa capacidade de generalização, uma vez que a mesma rede previu resultados para pontos geográficos diferentes no estado de Queensland.

Alguns aspectos do trabalho de Abbot e Marohasy são relevantes para o contexto deste trabalho. Primeiramente, cita-se a adoção de múltiplos parâmetros de entrada para as redes neurais, cujas influências na precipitação podem ocorrer de maneira diferenciada. Um outro aspecto relevante deste trabalho é a adoção bem sucedida de redes neurais com entradas defasadas no tempo para endereçar o problema da previsão de precipitação, que também será adotado neste contexto. Cita-se ainda a utilização de poucas camadas ocultas nas redes neurais, o que não afeta o poder de generalização e captura da não-linearidade.

Em (HTIKE; KHALIFA, 2010) modelos de predição de precipitação para a Malásia foram desenvolvidos usando Redes Neurais *Focused Time-Delay* (RNFTD) para realizar previsões “um passo à frente”. O conjunto de dados, consistindo de precipitação diária de janeiro de 1980 a maio de 2009, foi obtido do Departamento Meteorológico da Malásia e tratado para criar quatro conjunto de dados compostos por dados de precipitação mensal, trimestral, semestral e anual, todos com valores posteriormente normalizados no intervalo de -1 a 1.

Neste trabalho, modelos de RNFTD com uma camada escondida foram desenvolvidos para cada conjunto de dados nos quais o número ótimo aproximado de atrasos nas entradas e neurônios na camada escondida foram obtidos por um processo sistemático de tentativa e erro. Os autores usaram 80% do conjunto de dados para treinar e validar os modelos e os 20% restantes para testá-los, usando também o valor do MAPE obtido neste conjunto de dados para avaliar a acurácia dos modelos. O modelo de predição anual gerou o resultado mais preciso (94.25%), seguido dos modelos semestral, trimestral e mensal, de maneira respectiva.

Os autores conseguiram, portanto uma boa acurácia com as redes neurais selecionadas, sendo possível também observar a influência da variabilidade dos dados de entrada nos resultados fornecidos pelas redes, pois aquelas que utilizaram dados amostrados em menores espaços de tempos realizaram a tarefa de predição com menor precisão.

Considerando o trabalho de Htike e Khalifa é válido ressaltar a procedência dos dados empregados para treinar, validar e testar as redes neurais desenvolvidas os quais foram fornecidos por um departamento meteorológico. Uma solução análoga para a origem dos dados deste trabalho foi considerada. Também cabe observar a quantidade considerável de dados utilizados, de modo a prover bastante informação a ser assimilada pelas redes neurais. Outro fator que merece relevância é o tratamento prévio aplicado nos dados a serem empregados, normalizando-os em uma faixa de valores, o que diminue as flutuações durante o treinamento das redes, otimizando portanto seu desempenho.

Redes Neurais foram usadas em (SHUKLA et al., 2011) com o objetivo de prever o Índice Índiano de Chuva de Monções de Verão (ISMRI) usando índices de anomalias na Temperatura da Superfície do Mar (índices SST), especificamente o El Niño Oscilação Sul (ENSO). Os autores usaram o índice ISMRI correspondendo ao período de 1951 a 2003 bem como os índices Niño-1 + 2, Niño-3, Niño-3.4 e Niño-4 de 1950 a 2003 como parâmetros de entrada. A técnica de Análise de Componentes Principais foi usada para reduzir o número de parâmetros de entrada devido à quantidade relativamente pequena de dados considerados. Esta técnica gera um vetor de valores não-correlacionados que são chamados de Componentes Principais (PC, do inglês *Principal Components*), dos quais o primeiro, mais relevante, foi escolhido para ser usado como entrada da rede neural. Todos os dados foram, ainda, normalizados.

Redes Neurais de Múltiplas Camadas *Feed-Forward* com uma única camada escondida foram usadas por estes autores. Além disso, diferentes modelos de regressão foram construídos, em que cinco deles usaram os índices Niño individualmente, um usou todos os parâmetros de entrada e o último modelo usou somente o valor do componente principal. Estes modelos foram comparados de acordo com métricas de performance, tais como a raiz quadrada do erro médio quadrático, coeficiente de correlação e desvio padrão. Embora os modelos de regressão múltipla tenham

sido os melhores de sua classe, sua predição foi inferior à média do conjunto. Contudo, os modelos de redes neurais produziram resultados significantes, melhores que a média dos dados, o que reforçou a ideia de relacionamento não-linear entre o ISMIR e os índices Niño. Desta forma, os autores desfrutaram da capacidade de mapeamento não-linear das redes neurais, bem como demonstraram a importância da seleção de dados de entrada que forneçam informações importantes a respeito do problema analisado.

Do trabalho de Shukla et al. é interessante destacar a escolha de multiplas métricas de performance, uma vez que estas permitem avaliar o desempenho das redes neurais com diferentes perspectivas. A relação não-linear entre os dados de entrada e saída, e a consequente necessidade de capturá-la consiste, também, em um aspecto importante e, haja vista a notável capacidade das RNAs de captura de não-linearidades, utilizá-las para problemas deste domínio é uma escolha propícia. Além disso, nota-se a vital importância do pré-processamento dos dados para a geração de modelos de RNA eficazes, principalmente no que diz respeito a escolha das entradas empregadas nas redes.

Esta revisão da literatura destacou padrões comumente empregados no âmbito de trabalhos que endereçam problemas de predição de precipitação com a utilização redes neurais artificiais. No que diz respeito à quantidade de camadas escondidas, comumente opta-se por utilizar apenas uma, posto que seu dimensionamento, isto é, a decisão de quantos neurônios serão empregados na camada, é mais simples quando comparado ao dimensionamento de redes com duas ou mais camadas escondidas. Assim, ao invés de inferir mais parâmetros ajustáveis com diversas camadas, tende-se a apostar no correto dimensionamento de uma única camada escondida e na capacidade de generalização da rede resultante. Esta estratégia, que mostra-se consolidada na revisão de literatura realizada, também será empregada no âmbito deste trabalho.

Os trabalhos de previsão de séries temporais presentes na literatura frequentemente tratam de previsões um passo à frente, assim como no contexto do presente trabalho desenvolvido. Nestes trabalhos geralmente empregam-se redes neurais com características dinâmicas cujo desempenho em cenários de predição demonstrou-se bastante adequado. Em especial, as redes neurais com defasagens no tempo (em inglês, *time-delay*) são bastante adotadas para predição

de precipitações, produzindo resultados de acurácia satisfatória em diversos localidades ao redor do mundo, sendo por isso o modelo escolhido como referência para utilização neste trabalho.

Ao longo desta revisão, não foram encontrados trabalhos que tratem da predição de precipitação para a cidade de Manaus, Amazonas utilizando redes neurais. A superação desta lacuna motivou a realização deste trabalho.

# **Capítulo 3**

## **Previsão da Precipitação em Manaus, Amazonas**

Empregando os conceitos e técnicas anteriormente apresentados e considerando os trabalhos presentes na literatura que abordam a problemática de predição de séries temporais, foram desenvolvidos testes com o objetivo de gerar diversos modelos de redes neurais de modo a selecionar o mais apto para a realizar a tarefa de predição de precipitação para a cidade de Manaus.

Esta seção apresenta uma ambientação do cenário considerado, apresentando uma análise detalhada dos dados utilizados neste trabalho. Em seguida, discorre-se a respeito da metodologia concebida para conduzir a geração das redes neurais a serem utilizadas. Por fim, são apresentados as redes neurais obtidas nos experimentos, bem como seus resultados.

### **3.1 Caracterização do Cenário Considerado**

Este trabalho de conclusão de curso teve por objetivo a predição um passo à frente da ocorrência, ou não, de precipitação na cidade de Manaus. Esta é uma tarefa desafiadora, posto que a chuva em Manaus ocorre de maneira abundante e não-uniforme (SIOLI, 1991). Grande parte da chuva em Manaus é resultado da ação e influência de diversos sistemas de precipitação, dos quais citam-se a Zona de convergência intertropical, Anticiclones Subtropical, Alta da Bolívia,

Sistemas de Mesoescala Tropical, Zona de Convergência do Atlântico Sul, Sistemas de Escala Sinóptica, dentre outros (SILVA, 2012). Desta forma, nota-se que o fenômeno de precipitação de chuva em Manaus é resultado de um sistema naturalmente complexo.

De acordo com o sistema de classificação climático de Köppen, Manaus tem um clima de monções tropicais, com altas temperaturas consistentes durante o ano e chuva intensa de outubro a junho. Agosto é o mês mais seco desta região, com menos de 60mm de precipitação (ALAVARES et al., 2014). A precipitação em Manaus é um fator muito importante que atua diretamente no ajuste do clima na cidade (SIOLI, 1991). Isto ocorre pelo fato de a cidade de Manaus e seus arredores terem um clima naturalmente desconfortável, com calor e umidade de características equatoriais (SILVA, 2012).

Para este trabalho o conjunto de dados utilizado foi obtido a partir de medições realizadas por uma estação meteorológica automática implantada e mantida pelo Instituto Nacional de Meteorologia (INMET), ilustrada na Figura 3.1, localizada em 3°6'13.2552"S, 60°0'6596"W, 61.25m acima do nível do mar, entre os anos de 1970 a 2010 (INMET, 2016). Deste conjunto de dados, os valores ausentes ou corrompidos foram descartados resultando, desta forma em um conjunto de dados caracterizado por 11719 dias de dados válidos para as variáveis meteorológicas consideradas no trabalho aqui desenvolvido.



Figura 3.1: Estação meteorológica automática utilizada pelo INMET.

Fonte: [www.inmet.gov.br](http://www.inmet.gov.br)

## 3.2 Visão Geral da Base de Dados

Tendo em conta os dados oriundos da estação meteorológica resultantes da filtragem aplicada, foram consideradas as seguintes variáveis meteorológicas para a composição do conjunto de dados:

1. Temperatura Máxima Média (AHT, do inglês *Average High Temperature*);
2. Temperatura Mínima Média (ALT, do inglês *Average Low Temperature*);
3. Umidade Relativa Média (RH, do inglês *Relative Humidity*);
4. Velocidade Média do Vento (WS, do inglês *Wind Speed*);
5. Precipitação.

O atributo precipitação é o atributo de interesse considerado (atributo alvo), indicando não a quantidade de chuva, mas sim a ocorrência ou não de precipitação, de maneira binária. Os outros atributos são quantitativos e os valores de suas médias e desvios padrão são mostrados na Tabela 3.1.

Tabela 3.1: Média e desvio padrão das variáveis consideradas.

	<b>AHT</b>	<b>ALT</b>	<b>RH</b>	<b>WS</b>
<b>Média</b>	31.81	23.31	82.97	1.64
<b>Desvio Padrão</b>	2.17	1.20	7.84	1.60

Os atributos AHT e ALT, que indicam temperaturas em graus Celsius, têm sua distribuição mostrada em *boxplots* na Figura 3.2a. É possível notar que há certa sobreposição entre os valores, indicando que uma temperatura mínima pode vir a ser considerada como máxima, a depender de fatores sazonais. Também é possível observar, na Figura 3.2b, que não há uma clara separação para os valores AHT quando considera-se a ocorrência de chuva, observando-se o mesmo para os valores de ALT. Assim, estatisticamente não há como inferir a ocorrência ou não de precipitação para um determinado dia, analisando-se estritamente suas temperaturas.

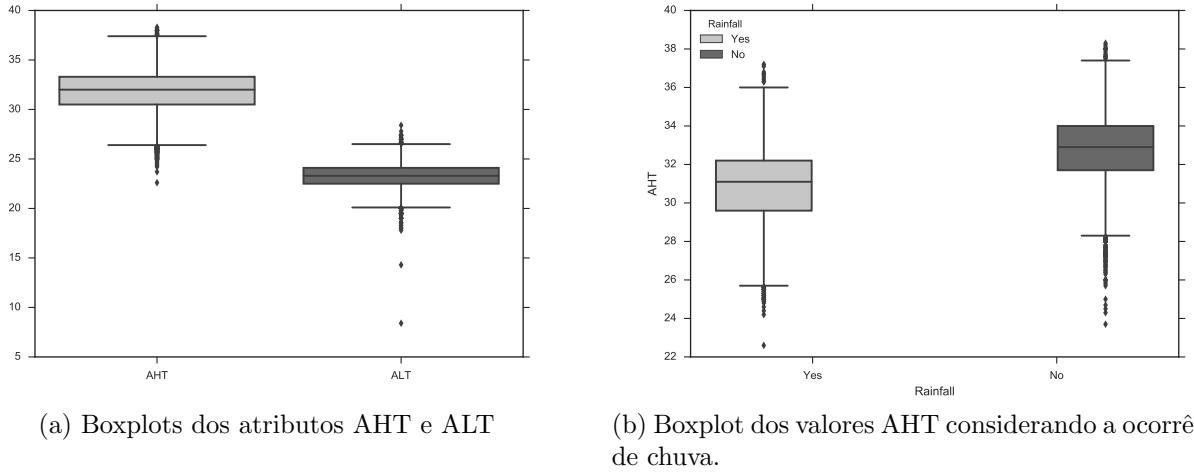


Figura 3.2: Boxplots obtidos a partir dos valores do conjunto de dados.

Com a finalidade de verificar quais atributos de entrada podem ter relação significante com a ocorrência de precipitação, obteve-se o coeficiente de correlação de Pearson desses atributos *versus* precipitação. Os resultados são mostrados na Tabela 3.2. É possível notar que há um correlação negativa significante para os atributos AHT, ALT e RH. No entanto, o coeficiente de correlação para WS e precipitação possui um valor despresível. Por esta razão, decidiu-se não utilizar o atributo WS nos passos seguintes da predição de precipitação.

Tabela 3.2: Coeficiente de correlação de Pearson dos atributos de entrada *versus* atributo de interesse.

	AHT	ALT	RH	WS
Precipitação	-0.45655336	-0.33254008	-0.33254008	-0.03047564

Dentre os 11719 dias de observação, foi possível constatar a ocorrência de chuva em 5830 dias (49.7%), e não ocorrência em 5889 dias (50.3%). A fim de verificar a possibilidade de detectar algum padrão estatístico na ocorrência de precipitação, efetuou-se um teste  $\chi^2$  (Qui-quadrado), considerando como hipótese nula que a precipitação está distribuída de acordo com uma variável com distribuição aleatória e uniforme. O valor  $p$  obtido foi de 0.58, indicando que a hipótese nula não pode ser rejeitada para um nível de significância de  $\alpha = 0.05$ . Isto significa que, para um nível de confiança de 95%, o padrão da ocorrência de precipitações em

Manaus parece ter natureza aleatória. Para ilustrar esta característica randômica observada, a Figura 3.3 exibe um gráfico os últimas 60 dias de chuva do conjunto de dados utilizado neste trabalho. Contudo, é interessante ressaltar que este resultado foi obtido considerado a precipitação isoladamente. Se outras variáveis forem levadas em conta, dadas as correlações identificadas anteriormente na Tabela 3.2, um padrão pode emergir, induzindo uma melhor performance na predição realizada.

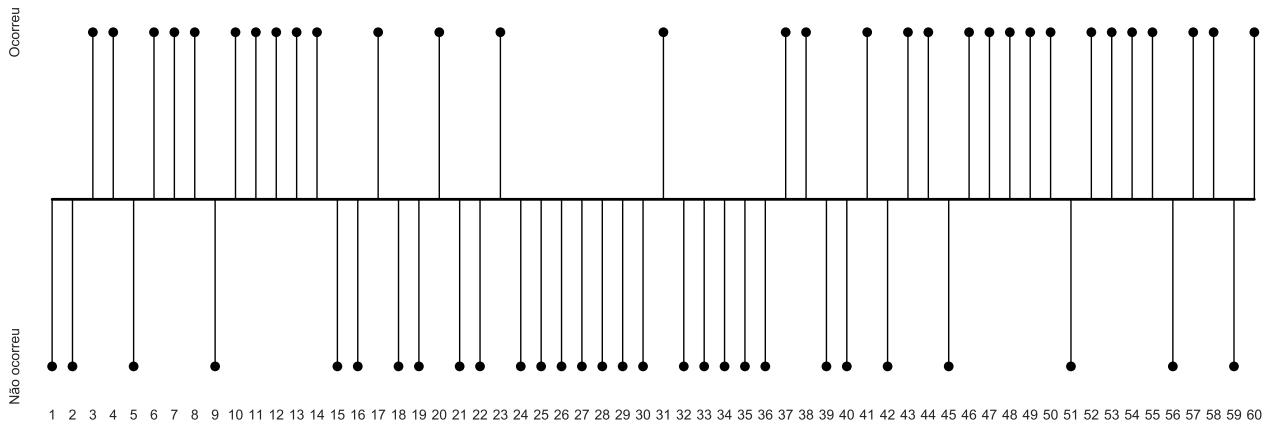


Figura 3.3: Gráfico de precipitação dos últimos 60 dias da base de dados. Fonte: Elaborada pelo autor.

### 3.3 Dimensionamento das Redes Neurais Artificiais

Conforme mencionado anteriormente, não há uma metodologia amplamente empregada para o dimensionamento de redes neurais nas tarefas de predição de precipitação. Assim, no escopo deste trabalho, foi desenvolvida a metodologia descrita a seguir, a qual especifica como elaborar, testar e treinar redes neurais para o cenário em questão. O problema considerado é o de predição um passo à frente da ocorrência de precipitação e os passos que compõem esta metodologia são:

1. **Tamanho da janela.** Escolher o tamanho da janela  $w$  a partir do conjunto de valores  $[1, 3, 5, 7, 10, 16, 21]$ ;
2. **Organizar dados de entrada.** O objetivo principal desta etapa é estruturar os dados a serem utilizados.

- (a) Organizar as  $w$  observações defasadas como dados de entrada: considerando os  $w$  valores anteriores de precipitação, AHT, ALT, RH;
  - (b) Adicionar aos dados de entrada os valores de AHT, ALT e RH do mesmo dia para o qual a precipitação será prevista;
  - (c) Dividir o conjunto de dados resultante em três subconjuntos considerando uma variação temporal, de modo que a janela de tempo deslize por estes subconjuntos que chegam a se sobrepor devido às amostras defasadas. A divisão foi feita de acordo com os seguintes intervalos:
    - i. Treinamento: 1970 - 2006;
    - ii. Validação: 2007 - 2009;
    - iii. Teste: 2010.
3. **Definir diversas arquiteturas de redes neurais.** Diferentes redes neurais *feedforward* serão consideradas para este cenário de predição.

- (a) **Neurônios de entrada ( $in$ ).** Definido de acordo com o número de parâmetros de entrada, em que  $in = 4 * w + 3$ , sendo  $w$  o tamanho da janela;
- (b) **Camada escondida.** Somente uma camada escondida será considerada;
- (c) **Neurônios na camada escondida ( $h$ ).** Definidos de acordo com a regra de Baum e Haussler (PALIT; POPOVIC, 2005) adaptada para este cenário:

$$h \leq \frac{N_{\text{TR}} \times E_{\text{TOL}}}{in + 1}, \quad (3.1)$$

em que  $N_{\text{TR}}$  é o número de exemplos de treinamento (Passo 2-c-i),  $E_{\text{TOL}}$  é a tolerância a erros definida como 0.01,  $in$  é o número de neurônios de entrada (Passo 3-a);

- (d) **Neurônios de saída.** Definidos como 1 por estar sendo considerada a predição um passo à frente.
4. **Treinar cada rede neural.** Considerando o conjunto de treinamento apropriado, de acordo com o tamaho da janela definido anteriormente;

5. **Validar cada rede neural.** Descartar redes neurais cujas acuráncias indiquem *overfitting* no conjunto de treinamento;
6. **Testar cada rede neural.** Registrar as performances e porcentagem de acertos para o conjunto de teste de cada rede (Passo 2-c-iii);
7. **Avaliação de Performance.** Determinar os valores do AIC, MSE e MAPE para cada rede obtida .

No último passo, após obter os valores do AIC, será possível ordenar as redes neurais de acordo com estes valores e selecionar o modelo que tem o mais baixo AIC, o qual indica sua melhor performance, de acordo com esta métrica, para o cenário considerado. Além disso, os valores do MSE e MAPE também serão obtidos, uma vez que são métricas consideradas mais intuitivas no que diz respeito à acurácia das redes neurais, sendo amplamente utilizados em trabalhos presentes na literatura.

### 3.4 Resultados

Em concordância com a metodologia estabelecida e apresentada na subseção anterior, foram obtidas 38 redes neurais, cujas arquiteturas estão listadas na Tabela 3.3, em que  $w$  é o tamanho da janela considerada para observações defasadas,  $in$  é o número de neurônios na camada de entrada,  $h$  é o número de neurônios na camada escondida e  $n$  é o número de dados para treinamento. Todos os neurônios da rede possuem função de ativação do tipo tangente hiperbólica sigmoidal, uma das mais comumente usadas em aplicações de redes neurais (MANDIC; CHAMBERS, 2001).

A partir dos resultados fornecidos pelas redes neurais ao processarem o conjunto de teste previamente designado, foram obtidos os valores das métricas Erro Médio Quadrático (MSE) e Erro Médio Percentual Absoluto (MAPE), conforme Equações 2.6 e 2.7, respectivamente. Ainda, empregando-se a Equação 2.9 e considerando as particularidades do cálculo do AIC mostradas na Seção 2.2.4, foram obtidos os valores desse critério para as redes neurais apresentadas, cujos resultados são mostrados na Tabela 3.4.

Tabela 3.3: Arquiteturas das redes neurais obtidas.

w	in	h	n	w	in	h	n
1	7	1	11718	3	15	7	11716
1	7	2	11718	5	23	1	11714
1	7	3	11718	5	23	2	11714
1	7	4	11718	5	23	3	11714
1	7	5	11718	5	23	4	11714
1	7	6	11718	5	23	5	11714
1	7	7	11718	7	31	1	11712
1	7	8	11718	7	31	2	11712
1	7	9	11718	7	31	3	11712
1	7	10	11718	7	31	4	11712
1	7	11	11718	10	43	1	11709
1	7	12	11718	10	43	2	11709
1	7	13	11718	10	43	3	11709
3	15	1	11716	13	55	1	11706
3	15	2	11716	13	55	2	11706
3	15	3	11716	16	67	1	11703
3	15	4	11716	16	67	2	11703
3	15	5	11716	21	87	1	11698
3	15	6	11716	21	87	2	11698

Tabela 3.4: Resultados para os valores de MSE, MAPE e AIC de cada arquitetura de rede neural obtida. O termo “Arqt.” denota a arquitetura das redes neurais.

w	Arqt.	MSE	MAPE	AIC	w	Arqt.	MSE	MAPE	AIC
1	7-1-1	0.1606	2.9911	-23517.63	3	15-7-1	0.1614	2.8826	<b>-24105.06</b>
1	7-2-1	0.1570	2.9090	-23681.25	5	23-1-1	0.1513	2.8893	-23901.20
1	7-3-1	0.1590	2.8663	-23431.63	5	23-2-1	0.1497	2.7620	-24007.33
1	7-4-1	0.1602	3.0250	-23645.02	5	23-3-1	0.1513	2.8893	-23805.20
1	7-5-1	0.1613	3.0356	-23718.90	5	23-4-1	0.1535	2.8135	-24005.93
1	7-6-1	0.1581	2.9637	-23603.01	5	23-5-1	0.1512	2.7526	-23877.10
1	7-7-1	0.1606	2.9600	-23750.20	7	31-1-1	0.1504	2.8796	-23928.21
1	7-8-1	0.1673	3.0587	-23431.62	7	31-2-1	0.1521	2.8591	-23953.78
1	7-9-1	0.1623	2.8997	-23897.44	7	31-3-1	0.1496	2.8069	-24031.17
1	7-10-1	0.1613	2.8997	-23772.80	7	31-4-1	0.1515	2.8455	-23889.85
1	7-11-1	0.1622	2.8874	-23726.92	10	43-1-1	0.1483	2.8711	-23934.75
1	7-12-1	0.1614	3.0567	-23562.86	10	43-2-1	0.1492	2.8558	-23904.03
1	7-13-1	0.1611	3.0332	-23625.87	10	43-3-1	0.1485	2.8356	-23863.45
3	15-1-1	0.1540	2.9149	-23811.52	13	55-1-1	0.1455	2.8531	-23980.03
3	15-2-1	0.1564	2.9151	-23850.41	13	55-2-1	0.1425	<b>2.7032</b>	-24046.14
3	15-3-1	0.1498	2.7936	-23942.28	16	67-1-1	0.1438	2.8448	-23988.27
3	15-4-1	0.1552	2.7894	-23976.73	16	67-2-1	0.1437	2.9010	-23911.69
3	15-5-1	0.1521	2.8525	-23887.08	21	87-1-1	<b>0.1422</b>	2.8297	-24003.03
3	15-6-1	0.1618	2.9722	-23945.01	21	87-2-1	<b>0.1422</b>	2.8388	-23832.25

### 3.5 Discussão

Analisando os valores de MSE presentes na Tabela 3.4, é possível notar um desempenho adequado e consistente das redes neurais obtidas, uma vez que todos os valores são menores que

0.2, uma grandeza considerada baixa para esta métrica. A Figura 3.4a apresenta um histograma com a distribuição obtida dos valores de MSE. Nesta figura, é possível notar que, apesar de flutuações, as grandezas produzidas estão, em sua maioria, concentradas nos valores em torno de 0.150 e 0.160. A Figura 3.4b, apresenta um *boxplot* dos valores desta métrica, no qual é possível identificar uma baixa variância.

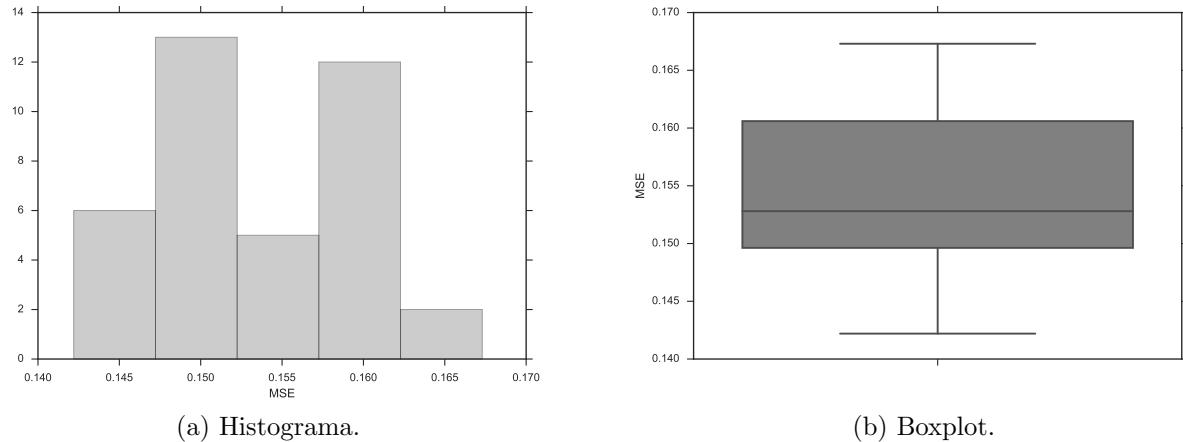


Figura 3.4: Histograma e boxplot para os valores resultantes do MSE.

Os menores valores de MSE na Tabela 3.4 estão associados às redes neurais que possuem maior tamanho de janela. Estes resultados indicam que tamanhos de janelas maiores podem induzir algum efeito de memória – em um sentido Markoviano – no processo preditivo, reduzindo, assim, os erros. Um gráfico do tamanho da janela *versus* o valor do MSE é mostrado na Figura 3.5, ilustrando tal relação observada nos resultados.

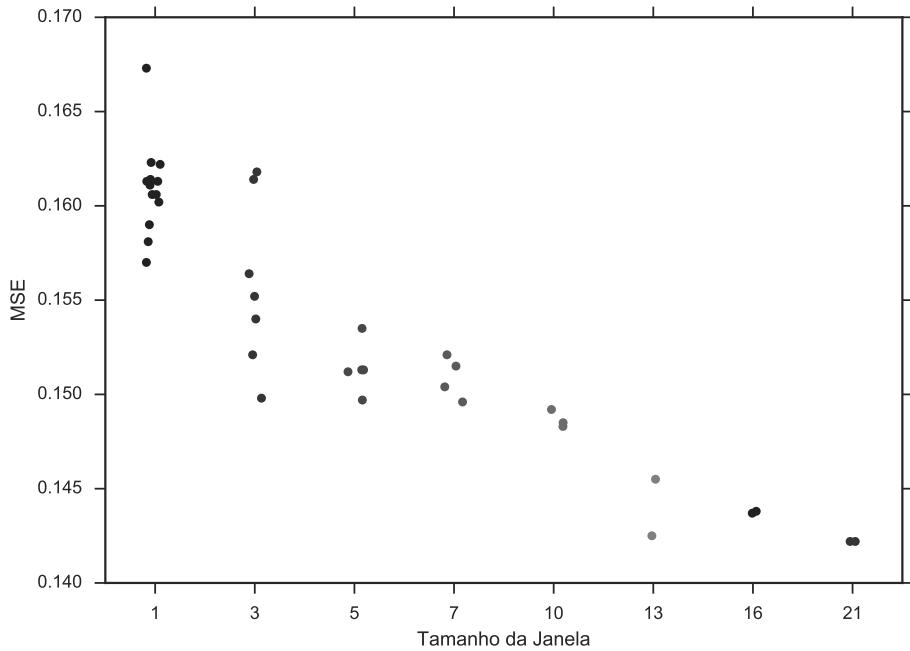


Figura 3.5: Tamanho da janela versus valor do MSE.

Ao relacionar a informação do tamanho da janela com o MSE, foi possível observar a relação inversamente proporcional entre estes valores nas redes neurais elaboradas. Esta relação vem a indicar que, quanto maior a quantidade de dias anteriores observados, melhor é o desempenho das redes neurais, em termos de MSE, que os tomam como entrada.

Com relação ao MAPE, que pode ser considerado como um método mais direto para a obtenção da acurácia dos modelos ( $\text{acurácia} = 1 - \text{MAPE}$ ), nota-se que os resultados são satisfatórios, todos em um fator de  $10^{-3}$ , podendo-se observar este fato mais claramente na Figura 3.6a. Além disso, também é possível notar que a maioria dos valores está concentrada na parte central do histograma. Assim como ocorrido com os valores do MSE, os valores do MAPE variam de maneira moderada em um intervalo de grandeza pequena, o que pode ser visto na Figura 3.6b.

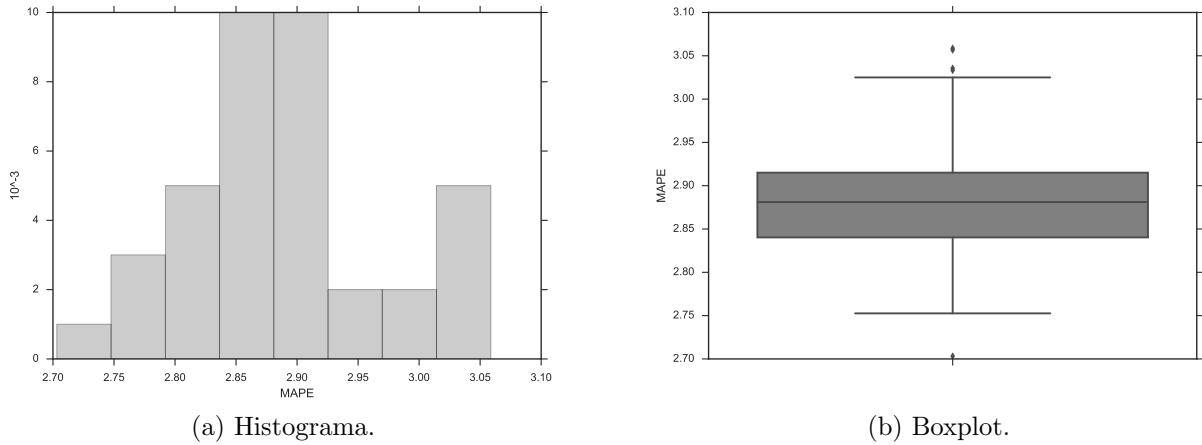


Figura 3.6: Histograma e boxplot para os valores resultantes do MAPE.

O melhor resultado observado para o MAPE foi o da rede neural com tamanho de janela 13 e arquitetura 55-2-1, cuja acurácia para o conjunto de teste foi de 99.72, o que indica que a diferença entre as previsões da rede neural e os valores reais (ocorrência ou não de precipitação) foi 99.72% favorável à rede neural desenvolvida. Todos os valores de MAPE indicam que a abordagem utilizada neste trabalho – com o uso de séries temporais e redes neurais para a predição de precipitação em Manaus – mostra-se adequada, provendo resultados certos a respeito da ocorrência de precipitação com uma alta porcentagem de respostas corretas.

Os valores AIC referentes às redes neurais resultantes estão também listados na Tabela 3.4. Assim como no caso do MSE e do MAPE, os valores desta métrica distribuíram-se em uma faixa de valores pequena, como pode-se ver nas Figuras 3.7a e 3.7b. Uma vez que o AIC tem por objetivo elencar um conjunto de modelos, de modo que quanto menor o AIC, mais adequado ele é considerado, pode-se observar que muitas das redes neurais propostas possuem desempenho satisfatório. Em particular, o melhor resultado foi observado na rede neural de arquitetura 15-7-1 com tamanho de janela 3, valor menor que o previamente observado para a rede com o melhor valor do MAPE, cujo tamanho de janela era 13. Vale ressaltar, porém, que o AIC mensura a qualidade da informação entregue pelas redes neurais, podendo-se considerar que o modelo destacado por esta métrica possui um melhor generalização, acarretando em um melhor desempenho em casos gerais, em contraste com o MAPE, que é obtido considerando

um cenário específico.

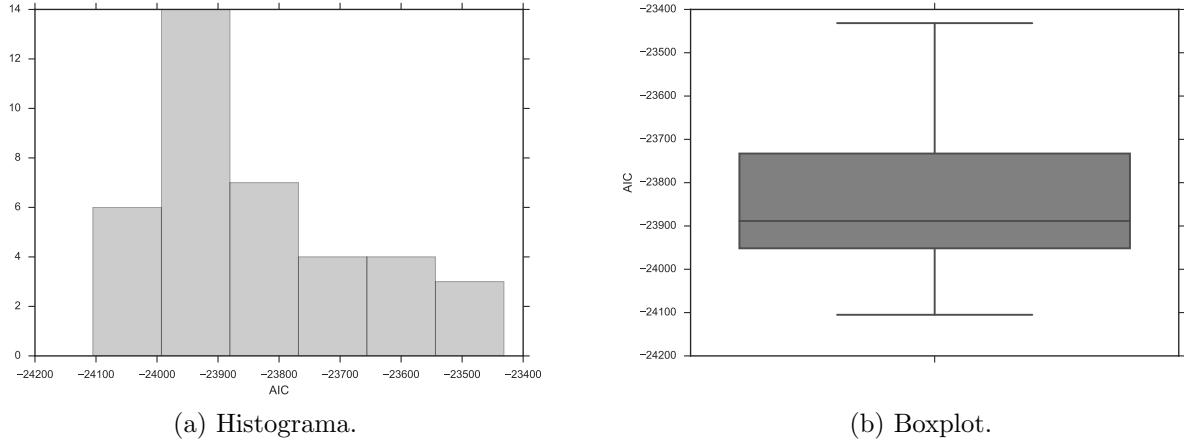


Figura 3.7: Histograma e boxplot para os valores resultantes do AIC.

Para sumarizar os resultados, a Figura 3.8a ilustra as previsões obtidas com a rede neural com maior MAPE, tamanho de janela 13 e arquitetura 55-2-1. A Figura 3.8b, por sua vez, mostra os resultados para a rede neural com melhor AIC, tamanho de janela 3 e arquitetura 15-7-1. Por fim, a Figura 3.8c exibe um comparativo entre estas redes, mostrando uma grande semelhança entre os resultados obtidos pelas mesmas.

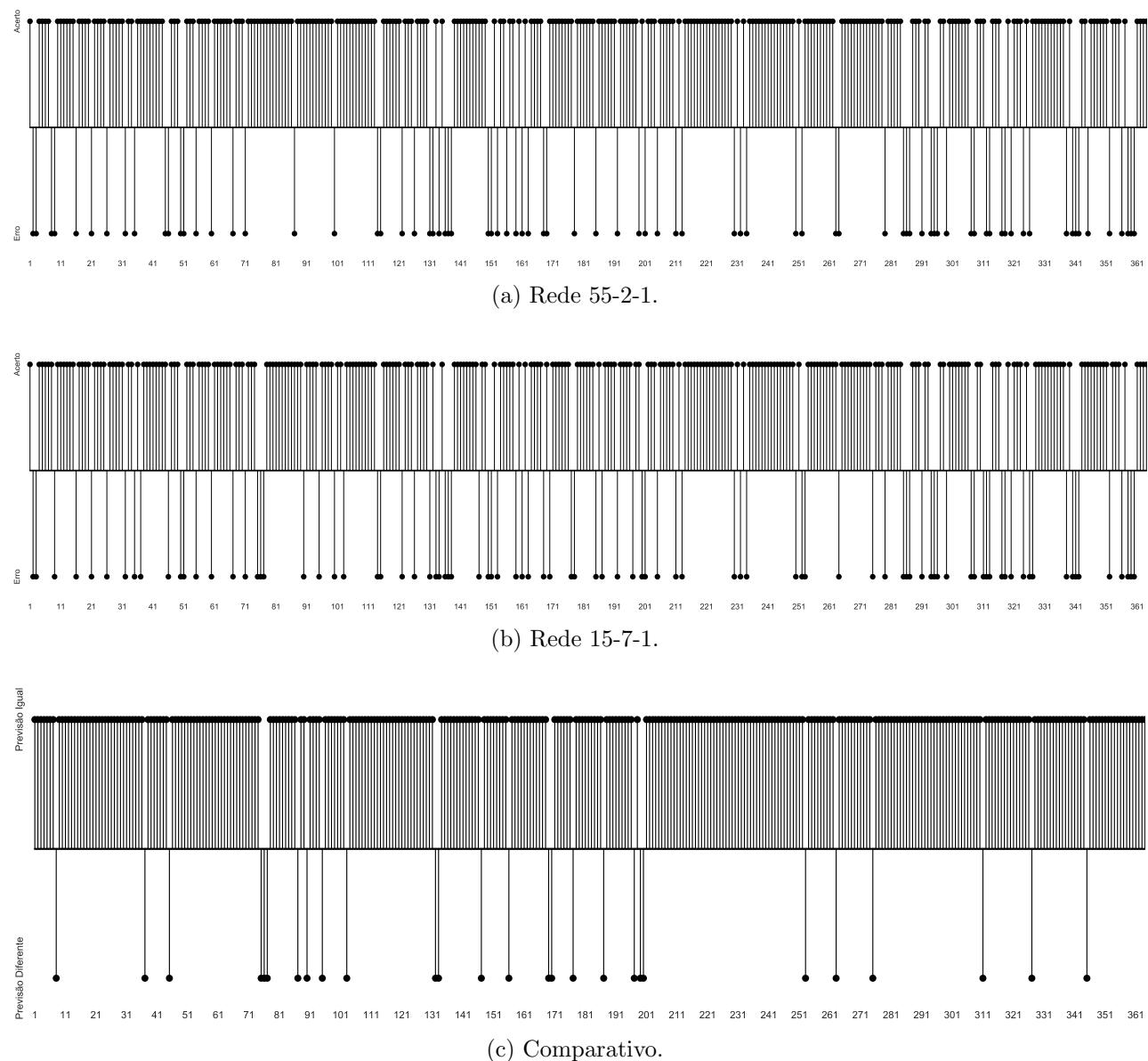


Figura 3.8: Resultados das melhores redes neurais observadas e um comparativo entre eles.

# Capítulo 4

## Considerações Finais

Neste capítulo é apresentada uma visão geral dos resultados e conclusões, as publicações obtidas ao longo do desenvolvimento deste trabalho de conclusão de curso e sugestões de trabalhos futuros.

### 4.1 Visão Geral dos Resultados e Conclusões

Este trabalho analisa o problema de predição da ocorrência de precipitação em Manaus, o qual foi endereçado por redes neurais artificiais utilizando uma abordagem de séries temporais. Para tanto, considerou-se dados oriundos de uma estação automática localizada na cidade e coletados ao longo de 40 anos.

Para endereçar este problema de maneira apropriada, foi preciso inicialmente conceber uma metodologia de dimensionamento das redes neurais. Com esta metodologia determinou-se a quantidade de neurônios em cada camada e como os dados foram organizados para serem apresentados à rede, pois utilizou-se redes neurais com entradas defasadas no tempo (*time-delay neural networks*) e diferentes tamanhos de janela para este atraso.

Com a metodologia considerada, obteve-se 38 diferentes redes neurais para o cenário em questão. Estas redes foram, então, treinadas, validadas e testadas. Destas redes neurais foram obtidos os valores do MSE e MAPE para os resultados das redes neurais ao processarem o conjunto de teste definido. Analisando estes valores, foi possível notar uma variação do MSE,

bem como a existencia de uma relação inversamente proporcional entre os valores desta métrica e o tamanho da janela utilizado em cada rede. Os valores de MAPE obtidos possuem grandeza satisfatória, com todos os valores em um fator de  $10^{-3}$  e variam em uma faixa de valores pequena, indicando uma acurácia estável e adequada para as redes obtidas. Desta métrica derivou-se a melhor topologia de rede neural obtida, com tamanho de janela 13 e arquitetura 55-2-1, a qual atingiu uma acurácia de 99.72%.

Os valores de AIC, também oriundos das saídas das redes obtidas ao processarem o conjunto de teste, foram analizados. Esta métrica permite avaliar a informação transmitida por um modelo, de modo a elencar o melhor. Da análise desta métrica foi obtido o modelo com arquitetura 15-7-1 com tamanho de janela 3.

Em síntese, este trabalho endereçou o problema da predição da precipitação em Manaus com redes neurais, obtendo modelos capazes de prever adequadamente o fenômeno em mais de 99.7% dos casos de teste considerados. Foi possível verificar que a utilização de variáveis meteorológicas coletadas ao longo de muitos anos permitiu a construção de modelos consistentes baseados na Inteligência Computacional e que há uma relação entre a previsão da precipitação com as características climáticas dos dias que a antecedem. Este trabalho ilustra uma aplicação interdisciplinar de conceitos da Computação e da Meteorologia.

Modelos estatísticos são tradicionalmente utilizados para a predição de variáveis climáticas, mas é conhecida a limitação destes modelos em incorporarem algumas complexidades inerentes ao processo que estão modelando. No caso da previsão de precipitação na cidade de Manaus, em particular, ainda não se conhece sequer um modelo estatístico para esta finalidade. Deste modo, a proposição de um modelo de rede neural artificial capaz de realizar a predição de precipitação para Manaus pode contribuir significativamente para as pesquisas em Meteorologia, proporcionando um melhor entendimento e motivando a realização de outros trabalhos.

## 4.2 Publicações Realizadas

Ao longo da consolidação deste trabalho, alguns resultados foram submetidos à conferências nacionais e internacionais. A listagem dessas publicações é mostrada a seguir:

1. LIMA, P. M.; GUEDES, E. B.; OLIVEIRA, M. B. L.; BoCliMa – Uma Ferramenta para Suporte ao Processamento de Dados de um Laboratório de Instrumentação Meteorológica. In: VII *Conferênciac Científica Internacional de la Universidad de Holguín*. Holguín, Cuba, 2015. v. 1. p. 1-10;
2. LIMA, P. M.; GUEDES, E. B.; Rainfall prediction for Manaus, Amazonas with artificial neural networks. In: 2015 *Latin America Congress on Computational Intelligence* (LACCI). Curitiba. 2015. p. 1-6;
3. LIMA, P. M.; GUEDES, E. B.; Uma Abordagem Baseada em Redes Neurais para a Predição de Chuva em Manaus, Amazonas. In: Congresso Amazônico de Computação e Sistemas Inteligentes. Manaus. 2015. v. 1. p. 149-158.
4. GUEDES, E. B.; LIMA, P. M.; OLIVEIRA, M. B. L.; *Neural Networks for Time Series Rainfall Forecasting: A Case Study in Manaus, Amazonas*. In: XIII Encontro Nacional de Inteligência Artificial e Computacional. Recife, Pernambuco, 2016. v. 1. p. 337-348;

A versão completa desses trabalhos encontra-se no Apêndice A.

### 4.3 Trabalhos Futuros

Para trabalhos futuros, pretende-se empregar outros modelos de Inteligência Computacional para endereçar o problema em questão, de modo a permitir comparações e a busca por eventuais aprimoramentos na metodologia de predição. Também pretende-se utilizar dados de outras estações meteorológicas localizadas em outras áreas de Manaus, com o objetivo de verificar como os resultados são generalizados considerando uma variação espacial.

O emprego de outras variáveis climáticas, bem como a utilização de técnicas estatísticas avançadas para uma seleção ainda mais criteriosa entre as variáveis disponíveis de modo a eleger as de maior influência neste fenômeno, pode colaborar para o desenvolvimento de um modelo mais robusto que melhor adeque-se à tarefa proposta neste trabalho.

Além disso, almeja-se avaliar a utilização da metodologia desenvolvida neste trabalho em outros cenários de precipitação, considerando outras localidades e tipos diferentes de clima.

# Referências Bibliográficas

- ABBOT, J.; MAROHASY, J. Application of artificial neural networks to rainfall forecasting in Queensland, Australia. *Advances in Atmospheric Sciences*, v. 29, n. 4, p. 717–730, 2012.
- ADAMS, C.; MURRIETA, R. S. S.; SANCHES, R. A. Agricultura e alimentação em populações ribeirinhas das várzeas do Amazonas: novas perspectivas. *Ambiente & Sociedade*, SciELO Brasil, v. 8, n. 1, p. 1–22, 2005.
- ALAVARES, C. A. et al. Köppen's climate classification map for Brazil. *Meteorologische Zeitschrift*, v. 22, n. 6, p. 711–728, 2014.
- BARRETO, J. M. *Introdução às Redes Neurais Artificiais*. 2002. ([https://intranet.dcc.ufba.br/pastas/mecateam/material\\_de\\_estudo/ia/iaconexionista/RedesNeurais.pdf](https://intranet.dcc.ufba.br/pastas/mecateam/material_de_estudo/ia/iaconexionista/RedesNeurais.pdf)).
- BEZDEK, J. C. On the relationship between neural networks, pattern recognition and intelligence. *International Journal of Approximated Reasoning*, n. 6, p. 85–107, 1992.
- BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. *Redes Neurais Artificiais - Teoria e Aplicações*. 2. ed. Rio de Janeiro: LTC, 2007.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, v. 2, p. 303–314, 1989.
- DARJI, M. P.; DABHI, V. K.; PRAJAPATI, H. B. Rainfall forecasting using neural network: A survey. In: *International Conference on Advances in Computer Engineering and Applications*. India: IMS Engineering College, 2015.
- DEMUTH, H.; BEALE, M.; HAGAN, M. *Neural Network Toolbox 6 – User's Guide*. [S.l.], 2009.
- ENGELBRECHT, A. P. *Computational Intelligence: an Introduction*. 2. ed. [S.l.]: John Wiley & Sons Ltd, 2007.
- FACELI, K. et al. *Inteligência Artificial: Uma abordagem de Aprendizado de Máquina*. 1. ed. [S.l.]: LTC, 2011.
- FRANK, R. J.; DAVEY, N.; HUNT, S. P. Time series prediction and neural networks. *Journal of Intelligent and Robotic Systems*, v. 31, p. 91–103, 2001.
- HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. [S.l.]: Pearson, 2009.
- HEBB, D. O. *The Organization of Behavior - A Neuropsychological Theory*. [S.l.]: Wiley and Sons, Inc., 1949.

- HTIKE, K. K.; KHALIFA, O. O. Rainfall forecasting models using focused time-delay neural networks. In: *2010 International Conference on Computer and Communication Engineering*. Malásia: IEEE, 2010. p. 1–6.
- HU, M. J. C. *Application of the ADALINE system to weather forecasting*. Dissertação (Mestrado) — Stanford El. Lab., Stanford, CA, 1964.
- INMET. *Instituto Nacional de Meteorologia INMET*. 2016. <<http://www.inmet.gov.br>>. Acessado em 20 de dezembro de 2016.
- KIPLI, K. et al. Performance of levenberg-marquardt backpropagation for full reference hybrid image quality metrics. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. [S.l.]: International Association of Engineers, 2012. v. 1.
- MANDIC, D. P.; CHAMBERS, J. A. *Recurrent Neural Networks for Prediction*. Inglaterra: Wiley, 2001.
- MATHWORKS, I. T. *Neural Network Toolbox*. 2016. <<http://www.mathworks.com/help/nnet/index.html>>. Acessado em 20 de dezembro de 2016.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MOREIRA, J. C.; SENE, E. de. *Geografia Geral e do Brasil*. [S.l.: s.n.], 2010.
- NAYAK, D. R.; MAHAPATRA, A.; MISHRA, P. A survey on rainfall prediction using artificial neural network. *International Journal of Computer Applications*, v. 72, n. 16, p. 32–40, 2013.
- OIKAWA, R. T.; ISHIKI, H. M. Modelos estatísticos e de redes neurais artificiais utilizados na predição de precipitação. *Fórum Ambiental da Alta Paulista*, v. 9, p. 19–34, 2013.
- PALIT, A. K.; POPOVIC, D. *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*. 1. ed. Londres: Springer, 2005.
- PANCHAL, G. et al. Searching most efficient neural network architecture using Akaike's Information Criterion (AIC). *International Journal of Computer Applications*, v. 1, n. 5, p. 41–44, 2010.
- ROJAS, R. *Neural Networks - A Systematic Introduction*. 2. ed. [S.l.]: Springer, 1998.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological Review*. [S.l.: s.n.], 1958. v. 65, p. 386–408.
- RUTKOWSKI, L. *Computational Intelligence – Methods and Techniques*. 1. ed. [S.l.]: Springer, 2008.
- SHUKLA, R. P. et al. Prediction of Indian summer monsoon rainfall using Niño indices: A neural network approach. *Atmospheric Research*, Elsevier, v. 102, p. 99–109, 2011.
- SILVA, D. A. da. Função da precipitação no conforto do clima urbano da cidade de Manaus. *Revista Geonorte*, v. 1, n. 5, p. 22–40, 2012.

SIOLI, H. *Amazônia: Fundamentos da Ecologia da maior região de florestas tropicais*. Manaus: Vozes, 1991.

TORRES, F. T. P.; MACHADO, P. J. de O. *Introdução à Climatologia*. [S.l.]: Cengage Learning, 2012.

WILAMOWSKI, J. D. I. B. M. *The Industrial Electronics Handbook. Second Edition: Intelligent Systems, Second Edition*. 2. ed. [S.l.]: CRC Press, 2011. v. 3. ISBN 9781439802830.

ZADEH, L. A. Soft computing and fuzzy logic. *IEEE Software*, v. 11, p. 48–56, 1994.

## **Apêndice A**

### **Artigos Publicados**

# **BOCLIMA – UMA FERRAMENTA PARA SUPORTE AO PROCESSAMENTO DE DADOS DE UM LABORATÓRIO DE INSTRUMENTAÇÃO METEOROLÓGICA**

## **BOCLIMA – A SOFTWARE TOOL TO SUPPORT DATA PROCESSING OF A METEOROLOGICAL INSTRUMENTATION LABORATORY**

*Patrick Magalhães de Lima<sup>1</sup>, Elloá B. Guedes<sup>2</sup>, Maria Betânia Leal de Oliveira<sup>3</sup>*

<sup>1</sup> Escola Superior de Tecnologia – Universidade do Estado do Amazonas, Brasil, patrickm.lima1@gmail.com,

<sup>2</sup> Escola Superior de Tecnologia – Universidade do Estado do Amazonas, Brasil, elloaguedes@gmail.com

<sup>3</sup> Escola Superior de Tecnologia – Universidade do Estado do Amazonas, Brasil, mabetanialeal@gmail.com

**ABSTRACT:** *This paper aims at presenting BoCliMa, a software tool developed to support the data processing of a Meteorological laboratory hosted at a Brazilian public university. BoCliMa is open-source and was developed using the Java platform, can export data generated to XML, has graphical interface to its users, and was build following the Model-View-Controller architecture. The input for BoCliMa is the data produced by a meteorological station and the output is the climatic data per day regarding temperature, pressure, humidity, index heat, among others. Such output is presented in a graphical way whose main purpose is to disclose meteorological bulletins to the population in general as well as to other researchers in the field.*

**Keywords:** *Applied Computing; Meteorological Data Processing; Climatic Bulletin.*

## **1. INTRODUÇÃO**

A Ciência da Computação torna-se cada vez mais necessária às outras ciências para auxiliar na construção e obtenção do conhecimento científico. A área da Meteorologia, em particular, faz uso de métodos, tecnologias e técnicas da Computação para auxiliar no processamento do grande volume de dados que dispõe, produzindo, por exemplo, previsões do tempo, análises de dados do clima, dentre outros. Os dados obtidos e produzidos são de interesse não só para pesquisadores desta área, como também para pesquisadores de outras áreas, autoridades governamentais e também para a população em geral, como é amplamente visto na mídia.

Os dados do clima costumam ser obtidos por meio da utilização de *estações meteorológicas*, as quais capturam informações sobre a temperatura, ocorrência de precipitações, umidade relativa do ar, dentre outros. Porém, embora tais estações adquiram dados essenciais, é somente após um processamento destes que estas informações tornam-se inteligíveis para a população em geral. Neste momento, esforços conjuntos entre profissionais da Meteorologia e da Computação são necessários, para saber como analisar os dados e como automatizar esta análise, gerando dados em tempo hábil sempre que possível.

Considerando esta necessidade de agregar conhecimentos da Computação para facilitar as tarefas de processamento de dados meteorológicos, este trabalho relata a construção do software BoCliMa, acrônimo para Boletim Climático de Manaus, desenvolvido para auxiliar na automatização da geração de boletins climáticos de um laboratório de instrumentação meteorológica. Este laboratório encontra-se sediado dentro de uma universidade pública brasileira, adquirido e construído com o auxílio de recursos governamentais para pesquisa e que tem como um de seus objetivos o fornecimento de dados meteorológicos para os órgãos governamentais, assim como, para a população em geral.

Para mostrar o software produzido, este artigo está organizado como segue. A contextualização da problemática e a motivação para o desenvolvimento do BoCliMa encontram-se descritos na Seção 2.1. A apresentação deste software, incluindo os dados do clima produzidos, as tecnologias utilizadas, uma visão geral de sua arquitetura e a apresentação deste considerando a perspectiva do usuário são ilustradas na Seção 2.2. Por fim, as considerações finais deste trabalho e as sugestões de trabalhos futuros são apresentadas na Seção 3.

## 2. CONTEÚDO

### 2.1 Contextualização problemática

O Laboratório de Instrumentação Meteorológica da Escola Superior de Tecnologia (EST) da Universidade do Estado do Amazonas (LAINSTRU-EST) encontra-se localizado nas dependências desta instituição de ensino ( $3^{\circ}5'32.5''S, 60^{\circ}0'59.69''W, 31m$  de altitude) e opera desde Março de 2010. Este laboratório é uma iniciativa do Projeto Rede Estadual de Meteorologia e Hidrologia do Amazonas, financiado com recurso da FINEP, e que tem como um de seus objetivos fornecer dados meteorológicos para os órgãos governamentais, assim como, para a população em geral, por meio de *boletins meteorológicos*.

Diferentemente da previsão do tempo, mais popularizada pela mídia, o boletim meteorológico tem por objetivo apresentar um diagnóstico das informações meteorológicas sobre um período de tempo, tais como, temperatura média, máxima e mínima, ocorrência de precipitações, umidade relativa, pressão atmosférica, dentre outras. Por meio da combinação de determinados dados meteorológicos é possível, inclusive, fornecer outras derivadas, tais como, o índice de calor, popularmente conhecido como sensação térmica.

No laboratório em questão, os dados para a geração destes boletins são obtidos a partir de uma *Estação Meteorológica Automática* localizada no Campus da Escola Superior de Tecnologia da Universidade do Estado do Amazonas. Esta estação encontra-se equipada com sensores para monitoramento da direção e velocidade do vento, pressão atmosférica, temperatura do ar, umidade relativa do ar, saldo de radiação, radiação solar incidente e refletida, radiação emitida pela superfície, radiação emitida pela atmosfera, radiação fotossinteticamente ativa, precipitação, perfil de temperatura no solo, fluxo de calor no solo, e conteúdo de água no solo.

Além das informações meteorológicas, a estação também fornece dados de controle, tais como carga da bateria, tempo estimado de funcionamento, etc. Todos os dados produzidos são persistidos em um arquivo-texto de maneira serializada a intervalos de tempo fixos e pré-determinados, usualmente de dez minutos. O arquivo, embora possua uma estrutura textual, assemelha-se a uma “tabela”, como maneira de organizar os dados. A Figura 1 ilustra um trecho deste arquivo.

```
1 'TIMESTAMP', 'RECORD', 'batt_volt_Min', 'PTemp', 'NRLite_Avg', 'CM3Up_Avg', 'CM3Dn_Avg', 'CG3UpCorr_Avg',
2 'CG3DnCorr_Avg', 'CNR1TC_Avg', 'CMA11Up_Avg', 'CMA11Dn_Avg', 'LI190S_Avg', 'VW_Avg', 'HFP01_Avg', 'STP01_50cm_Avg',
3 'STP01_20cm_Avg', 'STP01_10cm_Avg', 'STP01_5cm_Avg', 'STP01_2cm_Avg', 'CS106_Avg', 'HMP45C_Temp_Avg',
4 'HMP45C_RH_Avg', 'WindSpeed', 'WindDirection', 'TB4_Tot', 'TS', 'RN', '', 'W/m2', 'W/m2', 'W/m2', 'W/m2',
5 'deg_C', 'W/m2', 'W/m2', 'umol/mol', '%', 'W/m2', 'deg_C', 'deg_C', 'deg_C', 'deg_C', 'deg_C', 'hPa', 'dec_C', '%', 'm/s', 'Deg', 'mm',
6 ' ', 'Min', 'Smp', 'Avg', 'Avg',
7 'Avg', 'WVc:Averaged_Value', 'WVc:Averaged_Value', 'Tot'
8 '2014-02-05 09:40:00', 118726, 13.23, 28.78, 233.1908, 359.7406, 60.10703, 1249.177, 1290.342, 115.3246, 363.9763, 65.63671, 448.9503,
9 0.2546729, 'NAN', 29.45563, 28.85126, 28.97074, 29.36523, 29.92229, 1009.477, 27.5499, 79.00884, 0.7635838, 103.2079, 0
10 '2014-02-05 09:50:00', 118727, 13.24, 29.03, 218.5598, 342.636, 56.87075, 1259.051, 1300.909, 116.1448, 346.5667, 62.29135, 430.326,
11 0.2546285, 'NAN', 29.45018, 28.87249, 29.0521, 29.47018, 29.96688, 1009.529, 27.53578, 78.62649, 0.8266301, 113.4006, 0
```

Figura 1: Exemplo de um trecho dos dados encontrados no arquivo produzido pela estação meteorológica.

De acordo com a Figura 1, as linhas 1-7 mostram o cabeçalho do arquivo, que descreve quais valores serão persistidos a partir deste momento pela estação meteorológica. As linhas 8-9 e 10-11 exemplificam dois registros de dados da estação meteorológica feitos no dia 05 de Fevereiro de 2014 às 09h40min e às 09h50min, respectivamente.

Atualmente, os responsáveis pelo LAINSTRU-EST abrem este arquivo de maneira manual e, utilizando um software de planilha de dados, geram dados climatológicos de interesse, os quais serão utilizados para a construção de figuras com o resumo do boletim meteorológico. Tais figuras são geradas uma a uma com o auxílio

de um software para processamento de imagens. O resultado final é disponibilizado no site do laboratório para divulgação dos dados junto à comunidade em geral. Um exemplo do resultado deste processo é ilustrado na Figura 2.



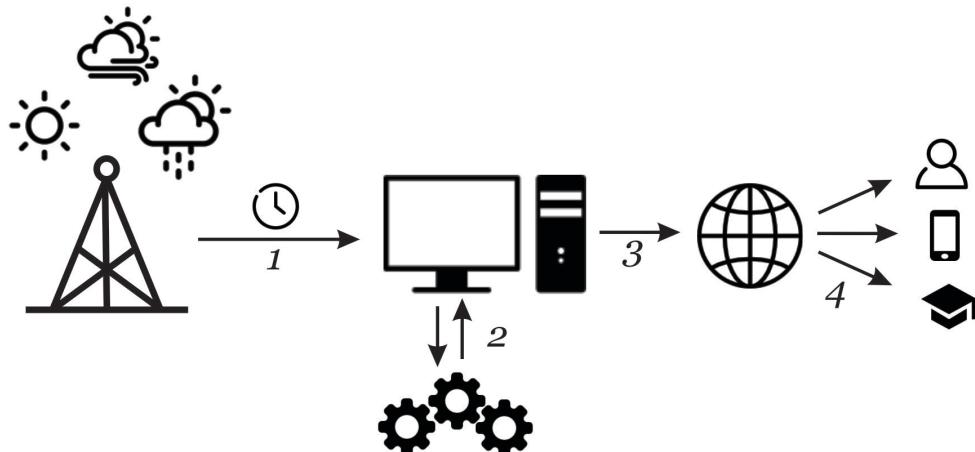
Figura 2: Exemplo de boletim meteorológico do mês de Outubro de 2014 produzido pelo LABINSTRU-EST

A geração de tais boletins como é feita atualmente, entretanto, está sujeita a alguns problemas. A seleção manual de dados e posterior processamento está muito propensa à imprecisão, pois o procedimento pode variar, ou à perda de dados, em virtude da grande quantidade e da difícil visualização da estrutura dos dados gerados pela estação meteorológica. Além destes fatores, incluem-se também a não-automatização de cálculos complexos, tais como para a obtenção do índice de calor, o dispêndio excessivo de tempo para geração dos dados e das figuras e a exigência de pessoal altamente treinado para realizar todas estas tarefas.

Levando em consideração a problemática identificada, este trabalho apresenta um software produzido para minimizar as dificuldades encontradas neste contexto: o *Boletim Climático de Manaus* (BoCliMa), que será apresentado em mais detalhes na seção a seguir.

## 2.2 BoCliMa – Boletim Climático de Manaus

O BoCliMa, acrônimo para Boletim Climático de Manaus, é um software desenvolvido especialmente para o Laboratório de Instrumentação Meteorológica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas. O principal objetivo resultante da utilização deste software é a geração automática de boletins meteorológicos da cidade de Manaus, Amazonas, Brasil para divulgação junto à comunidade em geral. A Figura 3 auxiliará na explicação do uso do BoCliMa no contexto do LABINSTRU-EST.



**Figura 3: Ilustração do contexto no qual o BoCliMa, representado pelas engrenagens, está inserido.**

A estação meteorológica, ilustrada à direita, coleta os parâmetros meteorológicos e os envia em intervalos periódicos para o servidor, conforme ilustra a seta 1. No servidor, um superusuário irá executar o BoCliMa sempre que desejar a fim de gerar um determinado boletim do clima, seja de um intervalo de tempo recente ou mais antigo. O BoCliMa produz os resultados especificados pelo superusuário no próprio servidor, conforme ilustra a seta 2. A partir daí, o superusuário disponibiliza os resultados na web, conforme seta 3, que podem ser utilizados, vide seta 4, pela comunidade em geral com caráter informativo, podem ser acessados via dispositivos móveis e podem até mesmo alimentar pesquisas acadêmicas.

### 2.2.1 Dados do Clima Produzidos

A partir dos dados produzidos pela estação meteorológica em um arquivo-texto, o BoCliMa gera dados de média, máximo e mínimo sobre oito aspectos do clima, listados a seguir:

- Pressão média;
- Temperatura mínima;
- Temperatura máxima;
- Umidade mínima;
- Umidade máxima;
- Velocidade máxima do vento e sua direção neste instante;
- Precipitação acumulada

Estes dados são categorizados de acordo com o turno de observação, podendo ser da madrugada, manhã, tarde ou noite conforme divisão ilustrada na Tabela I.

**Tabela I: Divisão de um dia em turnos, para categorização dos dados meteorológicos**

Período	Horário
Madrugada	00h01min – 06h00min
Manhã	06h01min – 12h00min
Tarde	12h00min – 18h00min
Noite	18h01min – 00h00min

Além destes dados, derivados de maneira direta a partir do processamento do arquivo produzido pela estação meteorológica, há também o cálculo do *índice de calor*, dado obtido a partir da umidade relativa do ar e da temperatura, combinados por meio de uma análise de regressão múltipla, conforme proposto por Rothfusz [1]. O cálculo do índice de calor é dado como segue:

$$\begin{aligned} \text{Index}_{\text{heat}} = & -42.379 + (2.04901523 \cdot T) + (10.14333127 \cdot rh) - (0.22475541 \cdot T \cdot rh) \\ & - (6.83783 \cdot 10^{-3} \cdot T^2) - (5.481717 \cdot 10^2 \cdot rh^2) + (1.22874 \cdot 10^{-3} \cdot T^2 \cdot rh) \quad (1) \\ & + (8.5282 \cdot 10^{-4} \cdot T \cdot rh^2) - (1.99 \cdot 10^{-6} \cdot T^2 \cdot rh^2) \end{aligned}$$

em que:

T: temperatura máxima do dia em graus Fahrenheit

rh: Umidade relativa do ar no instante da temperatura máxima

Há dois tipos de ajustes necessários no cálculo do índice de calor. Se  $rh < 13$  e a temperatura encontra-se entre 26.7 °C e 44.4 °C, é necessário subtrair do  $\text{Index}_{\text{heat}}$  o seguinte valor de ajuste:

$$\text{ADJUSTMENT} = [(13 - rh)/4] \cdot \sqrt{\frac{17 - \text{abs}(T - 95)}{17}} \quad (2)$$

em que:

abs: denota o valor absoluto

O outro tipo de ajuste ocorre se  $rh > 85\%$  e T encontra-se entre 26.7 °C e 30.6 °C. Neste caso, é necessário subtrair do  $\text{Index}_{\text{heat}}$  o valor de ajuste:

$$\text{ADJUSTMENT} = [(rh - 85)/10] \cdot [(87 - T)/5] \quad (3)$$

Assim, para cada dia de observação de dados meteorológicos, o BoCliMa obtém seu respectivo índice de calor. Este dado é muito importante, pois auxilia na tomada de decisão para evitar problemas de saúde pelo excesso de exposição ao calor, tais como cãimbras, esgotamento, dentre outros, podendo culminar em óbito [2].

## 2.2.2 Tecnologias Utilizadas e Arquitetura do Software

Para implementar o BoCliMa foi utilizada a plataforma Java, que consiste de uma linguagem de programação, bibliotecas e uma máquina virtual (*Java Virtual Machine*). Atualmente, Java encontra-se na versão 8 e é mantida pela empresa Oracle [3]. Esta plataforma foi escolhida para ser utilizada neste projeto por conter uma linguagem orientada a objeto, ser portável, dispor de uma grande quantidade de bibliotecas e ter uma comunidade de desenvolvimento bastante ativa.

A biblioteca Swing, integrante da plataforma Java, é voltada para construção de interfaces gráficas e geração de componentes visuais que proporcionem ao usuário uma melhor interação com o software produzido no que diz respeito à usabilidade [4]. Esta biblioteca foi utilizada para construção da interface gráfica do BoCliMa e também para a geração de figuras relativas aos resultados do processamento dos boletins meteorológicos, permitindo uma fácil visualização por parte dos usuários.

Por fim, a linguagem de marcação XML, foi utilizada para armazenamento dos dados processados [5]. Os dados são persistidos de acordo com uma *Definição de Tipo de Documento* (DTD), que consiste em uma gramática livre de contexto, descrevendo a estrutura geral que a persistência dos dados deve respeitar [6]. Como consequência, tem-se uma grande facilidade na portabilidade de dados, permitindo que o processamento produzido pelo BoCliMa possa ser incorporado por outros softwares e, assim, possa ser utilizado por outros pesquisadores interessados na área.

O BoCliMa foi construído seguindo o padrão arquitetural Model-View-Controller, amplamente utilizado para construção de aplicações que possuem interface com o usuário. De acordo com este padrão há três tipos de objetos: o modelo, o objeto da aplicação; a visão, apresentação na tela; e o controlador, que define a maneira como a interface do usuário reage às entradas do mesmo [7]. Além deste padrão arquitetural, os padrões de projeto Iterator, Facade e Singleton também foram adotados.

Para ilustrar a construção do software, a parte do modelo encontra-se ilustrada sob a forma de um diagrama de classes UML na Figura 4. No modelo, encontra-se todo o processamento responsável por transformar o arquivo de entrada gerado pela estação meteorológica em informações meteorológicas. A classe

FileManager abre o arquivo e segmenta-o em diversos objetos do tipo DataLine, que correspondem a cada linha no arquivo de entrada. As linhas são organizadas de acordo com o turno do dia, representado pela classe DayPeriod. Cada conjunto de quatro turnos compõe um WeatherDay e esta última faz uso da classe HeatIndex para obtenção do índice de calor. Há também uma classe responsável por auxiliar na persistência dos dados sob a forma de XML (classe SerializeWeatherDay) e uma classe responsável por atuar como interface de acesso ao modelo (classe WeatherDayFacade).

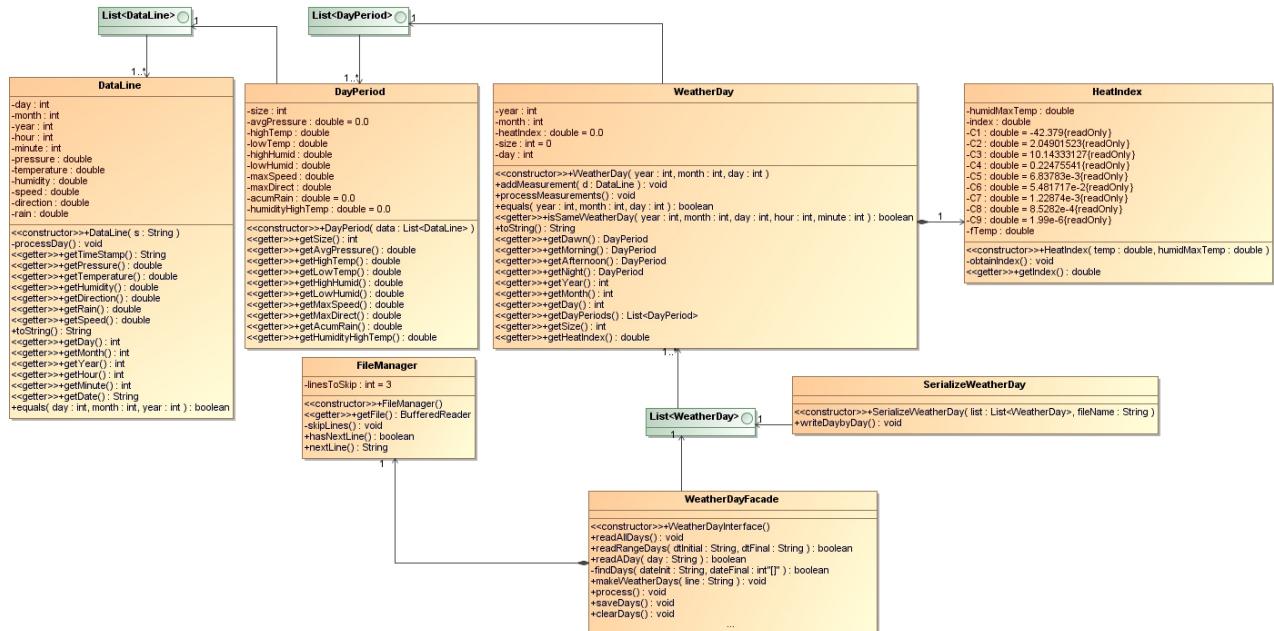
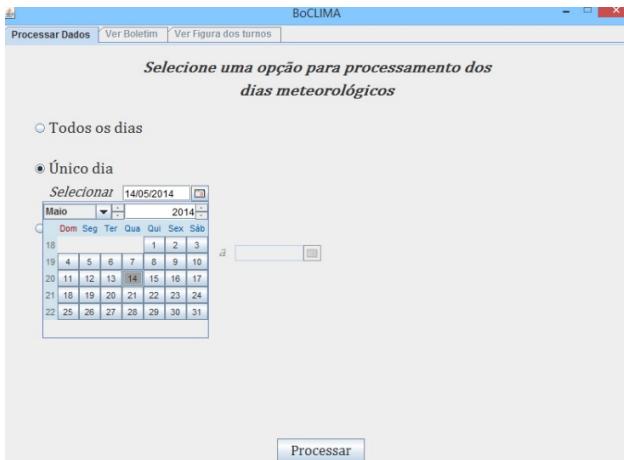


Figura 4: Diagrama de Classes UML representando o modelo do BoCliMa

Considerando as funcionalidades relatadas, o BoCliMa possui em sua versão atual um conjunto de 27 classes, organizadas segundo 8 pacotes, as quais somam 136 métodos e cerca de 1700 linhas de código.

### 2.2.3 Visão Geral do BoCliMa sob a Perspectiva do Usuário

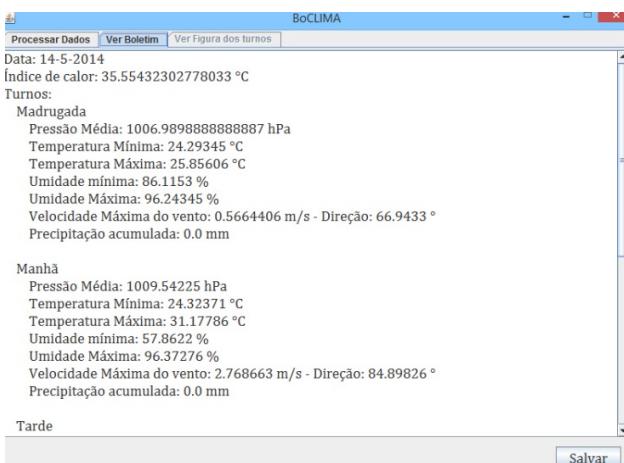
Nesta seção será ilustrada a utilização do BoCliMa sob a perspectiva do usuário, isto é, considerando capturas de tela da interação com o aplicativo. A tela inicial, ilustrada na Figura 5, permite a escolha do período de tempo a ser processado, podendo ser todo o período de dados, um dia específico ou um intervalo de tempo. Os resultados gerados para o período escolhido são ilustrados primeiramente via textual, conforme mostra a Figura 6, permitindo que sejam salvos sob a forma de um arquivo XML, conforme Figura 7. As imagens dos boletins climatológicos geradas pelo software são mostradas na Figura 8 e podem ser disponibilizados livremente pelos responsáveis pelo LABINSTRU-EST, para divulgação das informações junto ao público em geral.



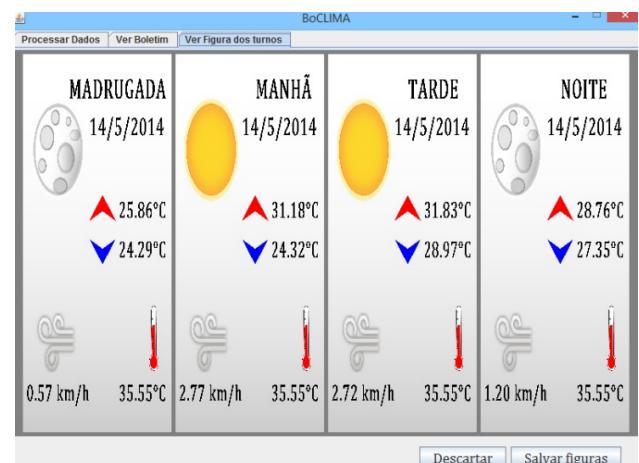
**Figura 5: Tela inicial**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<dados>
  <dia>
    <data>
      <ano>2014</ano>
      <mes></mes>
      <dia>14</dia>
    </data>
    <madrugada>
      <pressao_media>1006.989888888887</pressao_media>
      <temperatura_minima>24.29345</temperatura_minima>
      <temperatura_maxima>25.85606</temperatura_maxima>
      <umidade_minima>86.1153</umidade_minima>
      <umidade_maxima>96.24345</umidade_maxima>
      <velocidade_maxima>0.5664406</velocidade_maxima>
      <direcao_velocidade_maxima>66.9433</direcao_velocidade_maxima>
      <precipitacao_acumulada>0.0</precipitacao_acumulada>
    </madrugada>
    <manha>
      <pressao_media>1009.54225</pressao_media>
      <temperatura_minima>24.32371</temperatura_minima>
      <temperatura_maxima>31.17786</temperatura_maxima>
      <umidade_minima>57.8622</umidade_minima>
      <umidade_maxima>96.37276</umidade_maxima>
      <velocidade_maxima>2.768663</velocidade_maxima>
      <direcao_velocidade_maxima>84.89826</direcao_velocidade_maxima>
      <precipitacao_acumulada>0.0</precipitacao_acumulada>
    </manha>
    <tarde>
      <pressao_media>1005.441111111108</pressao_media>
      <temperatura_minima>28.97222</temperatura_minima>
      <temperatura_maxima>31.83194</temperatura_maxima>
    </tarde>
  </dia>
</dados>
```

**Figura 7: Arquivo XML gerado**



**Figura 6: Visualização dos dados de maneira textual**



**Figura 8: Exibição das imagens geradas**

Embora produzido diretamente para o LABINSTRU-EST, o BoCliMa pode ser obtido de maneira gratuita e livre, segundo a GPL 3.0 [8], por outros pesquisadores da área da Meteorologia que utilizem uma estação meteorológica similar. O fato de ser um software de código aberto também permite a implementação de melhorias por outros desenvolvedores e pesquisadores. O repositório do BoCliMa pode ser acessado em <https://code.google.com/p/boletim-tempo>, no qual encontra-se a versão mais atualizada do código-fonte, documentação sob a forma de Javadoc e versão executável, sob a forma de um jar.

### 3. CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado o software BoCliMa, voltado para o processamento de dados de uma estação meteorológica localizada na cidade de Manaus, Amazonas, Brasil, com o intuito principal de gerar boletins meteorológicos para divulgação junto à comunidade em geral. Além deste propósito, também é possível exportar os dados produzidos de tal modo que possam ser facilmente incorporados por outros aplicativos.

A motivação para a construção do BoCliMa partiu de um problema prático enfrentado por pesquisadores da área de Meteorologia, que processavam tais dados de maneira totalmente manual, em um processo demorado, exaustivo e passível de erros. A automatização desta atividade em virtude do uso do BoCliMa gera os mesmos resultados de maneira mais prática, rápida, totalmente automatizada e de fácil divulgação junto à população em geral, permitindo um maior alcance dos resultados produzidos junto ao público de interesse.

Do ponto de vista da Computação, considerando especialmente as práticas da Engenharia de Software, a construção deste software permitiu a utilização prática de diversos conceitos ligados ao desenvolvimento de software, incluindo a elicitação de requisitos, aspectos tecnológicos da utilização de uma linguagem de programação e suas bibliotecas, análise de aspectos de usabilidade, refatoramento, elaboração de um projeto arquitetural e adoção de padrões de projeto, até a documentação e deployment. Todos estes aspectos desenvolvidos num projeto de características essencialmente multidisciplinares.

Em trabalhos futuros almeja-se a captura de novos requisitos para o desenvolvimento de novas funcionalidades, melhoria nos aspectos de usabilidade, e a possibilidade de efetuar automaticamente a divulgação dos dados obtidos. Além disso, almeja-se verificar a possibilidade de utilizar elementos da inteligência computacional para previsão de dados meteorológicos.

### 4. AGRADECIMENTOS

Os autores agradecem ao Prof. Francis Wagner do Departamento de Meteorologia da Escola Superior de Tecnologia da Universidade do Estado do Amazonas pelas suas sugestões. O autor Patrick Magalhães agradece ao projeto Bolsa Samsung pelo apoio financeiro para realização deste trabalho.

### 5. REFERÊNCIAS

- [1] L. P. Rothfusz, "The heat index equation (or, more than you ever wanted to know about heat index)," NWS Southern Region Technical Attachment, Fort Worth, 1990.
- [2] T. L. da Silva e V. C. Almeida, "Influência do calor sobre a saúde e desempenho dos trabalhadores," em *Simpósio Maringáense de Engenharia de Produção*, Maringá, 2010.
- [3] P. Deitel e H. Deitel, Java – How To Program (Early Objects), 10th ed. ed., Prentice Hall, 2014.
- [4] M. Loy, R. Eckstein, D. Wood, J. Elliott e B. Cole, Java Swing, 2nd ed. ed., O'Reilly, Ed., 2003.
- [5] B. Evjen, K. Sharkey, T. Thangarathinam et al., Professional XML, Wrox, Ed., 2007.
- [6] J. E. Hopcroft, R. Montwani e J. D. Ullman, Introduction to Automata Theory, Languages and, 3rd ed. ed., P. Education, Ed., 2014.
- [7] E. Gamma, R. Helm, R. Jognson e J. Vlissides, Design Patterns: Elements of Reusable Object-, A.-W. Professional, Ed., 1994.
- [8] Free Software Foundation, "GNU general public license," 2007. [Online]. Available: <http://www.gnu.org/copyleft/gpl.html>.

# Rainfall Prediction for Manaus, Amazonas with Artificial Neural Networks

Patrick M. de Lima, Elloá B. Guedes

**Abstract**—The warm and humid Equatorial characteristics of Manaus, Amazonas as well as its geographic location and the topography gives the city a naturally uncomfortable weather. Because of that rainfall prediction in Manaus has an additional importance: it improves its citizens well-being. However, predicting rainfall in Manaus is a complex problem because it is under the influence of many precipitating systems. In this work, we addressed the problem of rainfall prediction in Manaus by using multilayer artificial neural networks. The input data was obtained from an automatic weather station during the years of 1970 to 2015. The performance factor considered was the normalized mean squared error. According to the results observed, a feedforward neural network with 2 hidden layers with 10 neurons each was the one that best addressed this problem. We also could see that recurrence in the neural networks did not improve the performance in the problem under consideration.

**Index Terms**—Rainfall Prediction; Artificial Neural Networks; Artificial Intelligence.

## I. INTRODUCTION

Climatological systems are characterized by complex modeling and having low predictability. In particular, due to the climatic changes, rainfall has been acquiring particular attention by researches given the occurrence of floods and droughts.

The capital of the Amazonas state, *Manaus* is the 7th most populous city in Brazil by number of habitants. One of the characteristics of Manaus is its weather, which is classified as tropical monsoon climate whose rainfall is very abundant but not uniform. For such characteristics, Manaus and its surroundings have a naturally uncomfortable weather, with warm and humid Equatorial characteristics. Given such weather characteristics, rainfall is important even to the well-being of its inhabitants. However, predicting it is a very hard task because the city suffers the influence of several precipitating systems [1].

Considering the importance of rainfall in this city and the hardness of predicting it, this work aims at showing an application of artificial neural networks in the rainfall prediction for Manaus. *Artificial neural networks* have the ability to perform distributed computation, to tolerate noisy inputs, and to learn [2]. Many works in the literature already show the potential of artificial neural networks for modeling meteorological forecasting [3].

To build such rainfall prediction model, we used meteorological parameters such as average high temperature, wind

speed, relative humidity, etc. collected from an automatic meteorologic station from 1970 to 2015. We submitted this data to different neural networks and could identify the best network which address the problem under consideration. Our results show that non-trivial neural networks can address such problem. By non-trivial we mean neural networks with multiple hidden layers and with several neurons in them.

To show our results, the present paper is organized as follows. A brief review of artificial neural networks is shown in Sec. II. We revisit other works in the literature which deal with rainfall prediction in Sec. III. The description of the problem under consideration in our work, the description of input and output data as well as the methodologies adopted are presented in Sec. IV. The results obtained are shown and discussed in Sec. V. Lastly, the final remarks and suggestion for future work are presented in Sec. VI.

## II. ARTIFICIAL NEURAL NETWORKS

Inspired in the human brain in which several physical neurons perform continuous parallel computations, the *Artificial Neural Networks* (ANN) are distributed parallel systems composed by simple processing units that compute some mathematical functions. To solve a problem using this computational model, we must first present some examples to the network. During this phase, called *training phase*, the network extracts relevant characteristics required to represent information. These characteristics are used afterwards to obtain answer to the problem proposed [4].

A *neuron* is an information-processing unit that is fundamental to the operation of a neural network. The simplest neuron provides a linear combination of  $N$  weights  $w_1, \dots, w_N$  and  $N$  inputs  $x_1, \dots, x_N$ , and passes the result through a nonlinearity  $f$  as shown in Fig. 1.

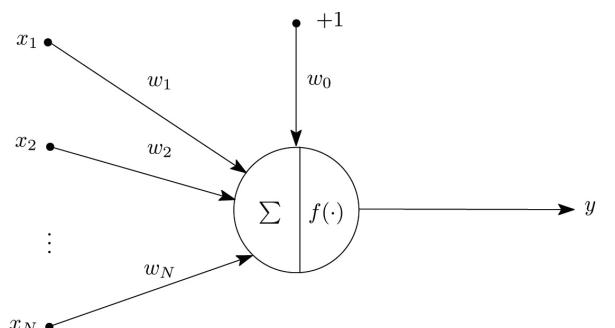


Figure 1: Representation of a neuron.

The input of a neuron is a vector  $\mathbf{x} = [x_1, \dots, x_N, 1]^T$ , whereas  $\mathbf{w} = [w_1, \dots, w_N, w_0]^T$  is referred to as the weight vector of a neuron. The weight  $w_0$  is the weight which corresponds to the bias input, which is typically set to unity. The sum operator performs a mapping on  $\mathbb{R}^n \rightarrow \mathbb{R}$ . The function  $f : \mathbb{R} \rightarrow (0, 1)$ , called *activation function*, is monotone and continuous, most commonly of a sigmoid shape [5]. The output  $y$  is obtained from the following equation:

$$y = f \left( \sum_{i=1}^N w_i \cdot x_i + w_0 \right). \quad (1)$$

In neural computation, such a neuron is referred to as the McCulloch-Pitts model, in recognition of the pioneering work done by McCulloch and Pitts [6]. In this model, the output of a neuron takes on the value of 1 if the induced local field of that neuron is nonnegative, and 0 otherwise. This statement describes the *all-or-none* property of the McCulloch-Pitts model [7].

Individual neurons have limited computational capacity. However, when they are connected in a network – *neural network* – they are capable to solve problems of higher complexity [4]. Models of neural networks are specified by the net topology, node characteristics and training or learning rules. From the perspective of connection patterns, neural networks can be grouped into two categories: feedforward networks, in which graphs have no loops, and recurrent networks, where loops occur because of feedback connections [5].

The *Multilayer Perceptron Neural Networks* (MLP) are those composed of several layers of neurons with sigmoidal activation function in the hidden layers. They can be used to approximate any continuous function [8]. A popular method for the training of MLPs is the *back-propagation algorithm* whose two phases are characterized as follows:

- 1) **Forward Phase.** In the forward phase, the weights of the network are fixed and the input signal is propagated through the network, layer by layer, until it reaches the output. In this phase, changes are confined to the activation potentials and outputs of the neurons in the network;
- 2) **Backward Phase.** In the backward phase, an error signal is produced by comparing the output of the network with a desired response. The resulting error signal is propagated through the network, again layer by layer, but this time the propagation is performed in the backward direction. In this second phase, successive adjustments are made to the synaptic weights of the network [7].

Neural networks can be applied to solve many problems regarding classification, categorization and forecasting. In classification, there are applications in image recognition, fraud detection, diagnostics, among others. In categorization, we can see its use in data mining, clustering techniques, DNA grouping, etc. Lastly, in forecasting, it is possible to use neural networks in weather forecasting, financial time series, dynamic systems modeling, among others [4].

### III. RELATED WORK

The adoption of artificial neural networks for meteorological forecasting was already adopted in many works in the literature, showing a relevant potential to this domain [3]. In this section, we describe some of the results in the literature regarding rainfall prediction with artificial neural networks.

The work of Abishek et al. consider the problem of rainfall prediction during the months of April to November in the state of Karnataka, India [9]. During this period, the months from May to October correspond to the monsoon season in which there are seasonal changes in atmospheric circulation and precipitation associated with the asymmetric heating of land and sea. The input data from April to November from 1960 to 2010 contains average humidity, wind speed, among others.

The authors considered three different multi-layer artificial neural network models: the feed forward back-propagation; the layer recurrent network and the cascaded back-propagation. They considered the mean square error obtained as a score for classification of the quality of the models adopted. With such considerations, the best results observed were those obtained with the back-propagation algorithm with a gradient descent with momentum weight and bias learning function. The number of neurons in the hidden layer for the best result was 10, but results obtained with 20 using this model of neural network were also very acceptable [9].

The work of Lee et al. considers the daily rainfall prediction at certain regions considering the daily rainfall at nearby areas in Switzerland [10]. First, they use a divide-and-conquer approach in order to separate the whole region into four sub-areas because they found out that the rainfall prediction in each area is modeled in a different way. It happened, for instance, because the pattern within smaller areas is continuous and smooth; and because the orographic effect exists, i.e., that climatological precipitation typically increases with elevation.

Regarding the rainfall prediction, in areas 1 and 4 they used the *radial basis function* (RBF) network. This kind of neural network provides an interpolating function which passes exactly through every data point. If there is noise, the interpolating function is smoother and averages over the noise to give the best generalization. Some modifications are also introduced to exact the interpolation procedure. In the neural networks built by the authors, the input data is the  $x$  and  $y$  coordinates of the region and the output is the rainfall prediction. The RBF architecture for region 1 was  $2 - 30 - 1$  while for region 4 was  $2 - 53 - 1$ . The root of mean square errors for these two areas were the smaller among the regions observed.

Considering areas 2 and 3, the authors used a linear regression instead of the neural network, because they considered the orographic effect. In practice, however, it apparently did not exist. When comparing the results obtained with RBF networks and the linear regression model, the predictions with the RBF networks turned out to be better because the error was smaller and because the area considered were bigger. According to the authors, the results are suitable for emergency

conditions, as well as long term management of contaminated regions [10].

Luk et al. [11] considered the problem of rainfall forecasting because of its importance for flood warning systems. The Upper Parramatta River catchment was the area under consideration, located in the western suburbs of Sydney, Australia. Their objective was to predict the forecast of rainfall one step ahead, i.e., 15 minutes before its occurrence. Firstly, they assumed that such rainfall is a Markovian process and that artificial neural networks can be used to represent such process.

They identified three configurations of neural networks that could be adequate to the problem under consideration: (i) multilayer feedforward networks; (ii) partial recurrent neural networks; and (iii) time delay neural networks. A partial recurrent neural network has a set of feedback connections which allow information to pass backwards, retaining information from previous time steps. So, the temporal structure of rainfall series is internally represented in this network by the feedback connections. Time delay neural networks are adequate for the recognition of local features within a larger pattern, independent of the positions of the local features. It is a feedforward network, but each layer is divided into several groups and then each group separately connected to the next layer [11].

The input data was the rainfall amounts during 15 minutes intervals at 16 rain gauges from January 1991 to September 1996. During this period, there were 34 storms and 1749 rainfall amounts at each site. The performance indicator was the normalized mean squared error.

Despite the particularities of the three models of neural networks used by these authors, they had comparable performance when developed and trained to reach their optimal complexities. For this storm, the networks made a good forecast of the decrease in rainfall rate in the storm evolution. However, none of the networks forecast a particular peak rainfall rate well. The results also suggested that the rainfall in this scenario has short-term memory, contributing to the understanding of this phenomenon [11].

#### IV. APPLICATION TO RAINFALL IN MANAUS, AMAZONAS

*Manaus* is located on the lower plateau of the Black river floodplain. It was founded around 1693 and is the capital city of the state of Amazonas in northern Brazil. Manaus is the 7th most populous city in the country, with a population of 2 million people in 2014 [12].

Manaus has a *tropical monsoon climate*, according to the Köppen climate classification system, with consistently high temperatures throughout the year and intense rainfall from October through to June [13]. Because the driest month, August, sees less than 60 mm of precipitation, the city's climate falls under the tropical monsoon climate category instead of the tropical rainforest climate category.

The city of Manaus and its surroundings have a naturally uncomfortable weather, with warm and humid Equatorial characteristics. It is due because two main factors: its geographic

location and the topography [1]. Sioli states that the rainfall in Manaus is very abundant, not uniform, and that it is a very important factor that shapes the weather in the city [14]. Most part of the rainfalls in Manaus are a result of many precipitating systems, such as the Intertropical convergence zone, Subtropical Anticyclones, Bolivian High, Tropical Mesoscale Systems, South Atlantic Convergence Zone, Synoptic Scale Systems, among others [1]. We can see that the rainfall in Manaus is the result of a very complex natural system.

##### A. Input and Output Data

The data considered was obtained from an *automatic weather station* which is composed by a central memory, called "data logger", connected to several sensors of meteorological parameters. The data is connected every minute and is available automatically within every hour. Such station is hosted and maintained by the *Instituto Nacional de Meteorologia* (INMET) located at  $3^{\circ}6'13.2552''S$ ,  $60^{\circ}0'6596''W$ , 61.25m above sea level [15]. Fig 2 shows the automatic weather station used by INMET to collect such data.



Figure 2: Automatic weather station used by INMET in Manaus.

From the INMET database, we obtained data from Manaus from January 1st, 1970 to January 1st, 2015. In this period, we retrieved 16436 days of data, but 3227 entries had to be discarded for having corrupted or incomplete data. We considered the following meteorological parameters per day of data observed:

- 1) Average High Temperature;
- 2) Average Low Temperature;
- 3) Average Relative Humidity;
- 4) Average Wind Speed;
- 5) Precipitation.

In our rainfall prediction model, parameters 1 to 4 will be the input data to the neural networks, while parameter 5 will be the output that we are trying to predict.

##### B. Neural Network Model

To build a rainfall prediction model to Manaus we considered the use of feedforward and recurrent multilayer perceptron neural networks. Multilayer networks are capable of performing just about any linear or nonlinear computation, and can approximate any reasonable function arbitrarily well

Table I: Multilayer neural network architectures and their training results.

Architecture	Transfer Function	Best Epoch	NMSE at Best Epoch	Early-Stopping Criterion
4-20-1	tansig	54	0.006559	Validation Checks
4-20-1	purelin	3	0.006482	Minimum Gradient Reached
4-50-1	tansig	12	0.006214	Validation Stop
4-50-1	purelin	4	0.006657	Minimum Gradient Reached
4-80-1	tansig	9	0.006666	Validation Stop
4-80-1	purelin	4	0.006761	Minimum Gradient Reached
4-10-10-1	tansig	12	0.005172	Validation Stop
4-10-10-1	purelin	3	0.006164	Minimum Gradient Reached
4-25-25-1	tansig	13	0.005567	Validation Stop
4-25-25-1	purelin	2	0.007142	Minimum Gradient Reached
4-40-40-1	tansig	10	0.006420	Validation Stop
4-40-40-1	purelin	2	0.005990	Minimum Gradient Reached
4-3-3-4-1	tansig	39	0.005575	Validation Stop
4-3-3-4-1	purelin	8	0.006706	Minimum Gradient Reached
4-8-8-9-1	tansig	20	0.005820	Validation Stop
4-8-8-9-1	purelin	5	0.007505	Minimum Gradient Reached
4-26-26-28	tansig	14	0.006465	Validation Stop
4-26-26-28-1	purelin	4	0.006047	Minimum Gradient Reached

Table II: Statistics of the NSME obtained grouped by number of hidden layers.

Number of Hidden Layers	Mean	Std. Deviation	Max	Min
1	0.006556	0.000193	0.006761	0.006214
2	0.006076	0.000685	0.007142	0.005172
3	0.006353	0.000700	0.007505	0.005575

Table III: Recurrent neural network architectures and their training results.

Architecture	Delays	Transfer Function	Best Epoch	NMSE at Best Epoch	Early-Stopping Criterion
4-10-10-1	2	tansig	50	0.005840	Validation Stop
4-10-10-1	3	tansig	33	0.006447	Validation Stop
4-10-10-1	4	tansig	22	0.006152	Validation Stop

[16]. In other similar works in the literature, some of them discussed in Section III, this model was successfully used in the problem of rainfall prediction. Furthermore, in the work of Luk et al. [11], the results obtained by more other kinds of neural networks were similar to those obtained by multilayer perceptron neural networks.

### C. Methodology

The Artificial Neural Network Toolbox from Matlab will be used to carry out the analysis on the weather data [16]. Out of the 13209 samples of input data, 70% will be used for *training* the neural network; 15% will be used for *validation*; and the remaining 15% will be used for *testing*. This division is well known in the literature.

The *performance measure* will be the *normalized mean square error* (NMSE) obtained during the test phase of the neural network under consideration. The NMSE is the sum of the square of the difference of the predicted value and the target value divided by the predicted value. With such a measure, it is possible to compare the performance of different neural networks.

We are going to consider different neural networks accord-

ing to three parameters:

- 1) **Number of neurons.** 20, 50, and 80 neurons will be considered. Previous results with smaller number of neurons were discarded due to a very high NMSE;
- 2) **Transfer function.** Tan-sigmoid (tansig) or linear (purelin);
- 3) **Number of hidden layers.** 1, 2, and 3 hidden layers were considered.

Sometimes a very low NMSE can be mistaken as good accuracy when in fact it points to an *overfitting problem*. Overfitting occurs when a neural network learns too many input-output examples, the network may end up memorizing the training data. It may do so by finding a feature (due to noise, for example) that is present in the training data, but not true of the underlying function that is to be modeled [7]. In order to avoid such problem, we are going to adopt *early-stopping criteria* in the method of training the neural networks:

- 1) **Validation Stop.** Validation vectors are used to stop training early if the network performance on the validation vectors fails to improve or remains the same for many epochs in a row;
- 2) **Maximum Mu Reached.** Mu, is a parameter of the

trainlm algorithm and measures the adapting/learning rate of the network. Maximum Mu reached means the learning rate has reached its maximum and further training will lead only to a validation stop or mostly, a minimum gradient;

- 3) **Minimum Gradient Reached.** Gradient is the direction of change of the values. Our algorithms implement gradient descent to find local minimum, and hence the term “minimum gradient” which is predefined for the network. [17]

Taking into account the data described and the methodology depicted in this section, we build the neural networks and tested their performance for the problem under consideration. In the next section we describe and analyze the results obtained.

## V. RESULTS AND DISCUSSION

By following the procedures described in the previous section, we started out by testing single hidden layer neural networks with 20, 50 and 80 neurons each. The transfer function between the internal layers was the tansig and then the purelin. The results observed are described in Table I. According to them, it can be seen that the results for such networks were very similar in terms of NMSE, whose standard deviation was 0.000193. In this category of neural networks, the best result was obtained by the neural network with 50 neurons in the hidden layer using the tansig transfer function. The neural network with 20 neurons in the hidden layer and tansig transfer function demanded a higher number of epochs when compared to the other models.

By increasing the number of hidden layers to 2, we tried to keep the number of neurons but reorganized them in the layers aiming at obtaining better results. In fact, when compared to the single layer neural networks, the average NMSE was 0.006076 while the previous architecture average NMSE was 0.006556. The standard deviation was a little higher than for single layer neural networks. It may indicate us that the number of neurons in the hidden layers is a more important factor for the performance of this kind of architecture.

Lastly, for neural networks with 3 hidden layers, but on average the same number of neurons considered for 1 and 2 layers, the average NMSE obtained was 0.006353. The standard deviation of NMSE was also the higher when compared to the neural networks with one and two hidden layers. As we can see, the results were not better than for 2 hidden layers.

Statistics of the NMSE of the neural networks considered so far are described in Table II. For the feedforward neural networks, we can see that the architecture with 2 hidden layers had the best performance result (mean and minimum NMSE), seeming to capture the patterns of rainfall prediction much more efficiently when compared to the other two types considered. The maximum NMSE can be seen in neural networks with 3 hidden layers, what may indicate that an increase in the number of layers is not a decisive factor for performance increase.

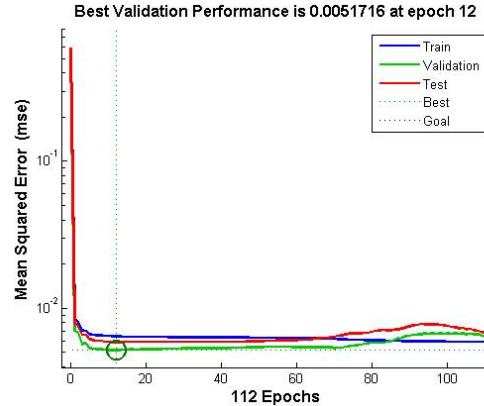


Figure 3: Performance result for the  $4 - 10 - 10 - 1$  neural network. The  $x$  axis indicates the number of epochs and the  $y$  axis indicates the mean squared error.

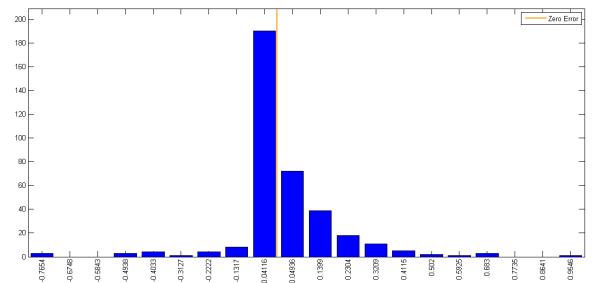


Figure 4: Error histogram for the year of 1996 obtained by the  $4 - 10 - 10 - 1$  neural network. In the  $x$  axis we have the errors, grouped in 20 bins, and in the  $y$  axis we have their instances.

The best overall result was obtained by the feedforward  $4 - 10 - 10 - 1$  neural network that reached the lower NMSE (NMSE = 0.005172) with 12 epochs and tansig transfer function. As it can be observed, the results for 20 neurons in single layer and three hidden layers networks were considerably high, showing us that it is not a mere question of how much neurons to use, but also how to organize them.

Taking a closer look at the  $4 - 10 - 10 - 1$  neural network, its performance is shown in Fig 3, where the best epoch observed, 12, is highlighted. We can see that train, test and validation results seems to converge in the best epoch and also after it.

We choose the year of 1996 to illustrate an histogram of errors, depicted in Fig 4. We can see a normal distribution tendency in it. This result was also observed in other years, such as in 1995, 1999, 2005, 2006, 2009, 2012, 2013 and 2014. All these years are in the test group.

By plotting a comparison between observed values and predicted values by the  $4 - 10 - 10 - 1$  neural network, we have the results shown in Fig. 5. Although these results are the best obtained, this neural network still seems not to capture with precision some days with a very high rainfall.

Given the problem observed, we tried to add recurrence

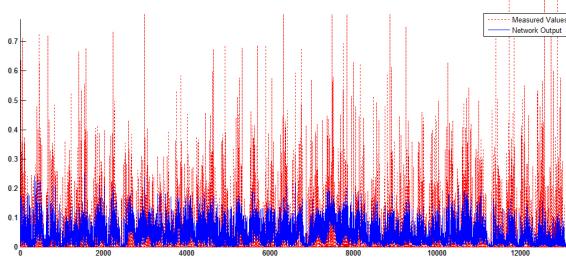


Figure 5: Observed versus predicted results for the  $4 - 10 - 10 - 1$  neural network. In the  $x$  axis we have the days observed while in the  $y$  axis we have the normalized rainfall values.

in the  $4 - 10 - 10 - 1$  neural network trying to see if it would capture such phenomenon with more precision. The results for the  $4 - 10 - 10 - 1$  neural networks with 2, 3 and 4 delays can be observed in Table III. According to the performance factor previously chosen, no improvement of performance was observed by capturing recurrence in the problem under investigation. We can see that the feedforward neural network  $4 - 10 - 10 - 1$  with tansig transfer function is the best model to the rainfall prediction problem in Manaus.

## VI. FINAL REMARKS

In this work we considered the problem of rainfall prediction in Manaus, Amazonas. Rainfall in Manaus is very important even to the well-being of its citizens due to the warm and humid Equatorial characteristics which causes a naturally uncomfortable weather. We addressed this problem using multilayer artificial neural networks because they can approximate any reasonable function arbitrarily well.

We used data from 1970 to 2015 and different neural networks architectures. According to the results observed, the neural network which bests addressed the problem consideration is the feedforward  $4 - 10 - 10 - 1$  artificial neural network with tansig transfer function in the hidden layers. This neural network normalized mean squared error was 0.005172 and the best epoch was 12. By trying to advance the results with recurrent artificial neural networks with the same number of neurons and hidden layers, no such advances could be observed.

The problem considered and its solution is very important not only to the computer intelligence domain – in which neural network is a good model for the solution. It can provide insights to meteorologists to improve the understanding of the rainfall in this very important region of the planet, surrounded by the Amazon rainforest. Besides, to public authorities, we hope that the results of this work can lead to better strategies to protect the population of Manaus from natural disasters caused by the rain.

In future works, we still would like to investigate the problem of rainfall prediction in Manaus in more details. We would like to separate the data according to the seasons of the year, trying to capture the rainfall season more particularly. Besides, other artificial neural networks needed to be tested,

such as the time series neural networks. This model was successfully adopted in the rainfall prediction problem in Australia [11].

## ACKNOWLEDGMENTS

The authors would like to thank Profs. Maria Betânia Leal, Francis Wagner and José Augusto Veiga from the Meteorology Department of our university. We also would like to thank Prof. Ernande Melo for the advices regarding artificial neural networks. Patrick Magalhães de Lima would like to thank Samsung for the financial scholarship provided during this work.

## REFERENCES

- [1] D. A. da Silva, “Função da precipitação no conforto do clima urbano da cidade de manaus,” *Revista Geonorte*, vol. 1, no. 5, pp. 22–40, 2012.
- [2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Pearson Education, 2003.
- [3] R. T. Oikawa and H. M. Ishiki, “Modelos estatísticos e de redes neurais artificiais utilizados na predição de precipitação,” *Fórum Ambiental de Alta Paulista*, vol. 9, no. 8, pp. 19–34, 2009.
- [4] A. de Pádua Braga, A. P. de Leon F. de Carvalho, and T. B. Ludermir, *Redes Neurais Artificiais: Teoria e Aplicações*, 2nd ed., LTC, Ed., 2007.
- [5] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction*, Wiley, Ed., 2001.
- [6] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [7] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Prentice-Hall, Ed., 2009.
- [8] R. Duda, P. Hard, and D. Stork, *Pattern Classification*, Wiley, Ed., 2001.
- [9] K. Abhishek, A. Kumar, R. Ranjan, and S. Kumar, “A rainfall prediction model using artificial neural network,” in *IEEE Control and System Graduate Research Colloquium*, 2012.
- [10] S. Lee, S. Cho, and P. M. Wong, “Rainfall prediction using artificial neural networks,” *Journal of Geographic Information and Decision Analysis*, vol. 2, no. 2, pp. 233–242, 1998.
- [11] K. C. Luk, J. E. Ball, and A. Sharma, “An application of artificial neural networks for rainfall forecasting,” *Mathematical and Computing Modelling*, vol. 33, pp. 683–693, 2001.
- [12] J. C. Moreira and E. de Sene, *Geografia Geral e do Brasil*, E. Scipione, Ed., 2010.
- [13] C. A. Alavares, J. L. Stape, P. C. Sentelhas, J. L. M. Gonçalves, and G. Sparovek, “Koppen’s climate classification map for brazil,” *Meteorologische Zeitschrift*, vol. 22, no. 6, pp. 711–728, 2014.
- [14] H. Sioli, *Amazônia: Fundamentos da ecologia da maior região de florestas tropicais*, Vozes, Ed. Vozes, 1991.
- [15] INMET, “Instituto nacional de meteorologia,” <http://www.inmet.gov.br>, 2015.
- [16] H. Demuth, M. Beale, and M. Hagan, “Neural network toolbox – user’s guide,” The Mathworks, 2009.
- [17] K. Abhishek, M.P.Singh, S. Ghosh, and A. Anand, “Weather forecasting model using artificial neural network,” *Procedia Technology*, vol. 4, pp. 311–318, 2012.

# **Uma Abordagem Baseada em Redes Neurais para a Predição de Chuva em Manaus, Amazonas**

**Patrick Magalhães de Lima, Elloá B. Guedes**

<sup>1</sup>Núcleo de Computação  
Escola Superior de Tecnologia  
Universidade do Estado do Amazonas  
Av. Darcy Vargas, 1200 – Manaus – Amazonas

{pml.eng, ebgcosta}@uea.edu.br

**Abstract.** *The weather in Manaus, Amazonas is typically Equatorial warm and humid which causes a natural discomfort for its inhabitants. One of the main agents for the local weather regulation is the precipitations, whose prediction is very important although difficult, because the weather in the city is under the influence of many precipitation systems. This work aims at presenting an approach based on multilayered artificial neural networks to predict the occurrence of rainfall in Manaus. Data from an automatic meteorological station from 1970 to 2014 was used. After testing a set of 2500 different neural network that address this problem, it was possible to identify the network with 4 – 9 – 7 – 1 architecture with best performance, being able to predict rainfall with almost 70% certainty.*

**Resumo.** *O clima na cidade de Manaus, Amazonas é caracterizado por calor e umidade tipicamente equatoriais, responsáveis por um desconforto natural a seus habitantes. Um dos principais agentes na regulação do clima local é a precipitação, sendo sua predição de apreciável importância, tratando-se porém, de um procedimento difícil, pois o clima da cidade é influenciado por diversos sistemas de precipitação. Este trabalho apresenta uma abordagem utilizando redes neurais artificiais de múltiplas camadas para prever a ocorrência de precipitações na cidade de Manaus. Foram utilizados dados climáticos de 1970 a 2014, provenientes de uma estação meteorológica automática. Após testar um conjunto de 2500 redes neurais que endereçam este problema, foi possível identificar a rede com arquitetura 4 – 9 – 7 – 1 com melhor desempenho para este problema, capaz de prever precipitações com cerca de 70% de acerto.*

## **1. Introdução**

Um dos fatores importantes para a sensação de conforto na cidade de Manaus, Amazonas é a ocorrência de precipitações, pois o clima nesta cidade é caracterizado pelo calor e pela umidade. As precipitações em Manaus ocorrem de maneira mais abundante entre os meses de Outubro a Julho e, segundo Sioli [Sioli 1991], a chuva em Manaus é abundante e não uniforme.

A ocorrência de precipitações em Manaus é resultado da influência de diversos sistemas de precipitação, a exemplo da Zona de Convergência Intertropical, Alta da Bolívia, Zona de Convergência do Atlântico Sul, dentre outros [Alvares et al. 2014, da Silva 2012]. Levando isto em consideração, é possível afirmar que a ocorrência de precipitações em Manaus é um fenômeno complexo e difícil de prever.

A previsão de precipitações em diversos lugares do mundo é um problema bastante endereçado pelas *Redes Neurais Artificiais* (RNAs). RNAs são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas. Estas redes são muito utilizadas em problemas de classificação, categorização e previsão, tais como reconhecimento de imagens, detecção de fraudes, mineração de dados, dentre diversos outros [de Pádua Braga et al. 2007].

Este modelo de computação é bastante utilizado neste tipo de problema por várias razões, a citar: (i) RNAs são dirigidas por dados (*data driven*) e não demandam pré-requisitos restringentes sobre o que está sendo modelado; (ii) RNAs podem predizer padrões que não são fornecidos durante o treinamento, isto é, são capazes de generalizar; (iii) RNAs são eficientes no treinamento de grandes amostras de dados graças a sua capacidade de processar em paralelo; (iv) RNAs possuem a habilidade de detectar relações complexas e não-lineares entre as variáveis dependentes e independentes [Darji et al. 2015].

Considerando a importância das RNAs no contexto de previsão de precipitações, este trabalho apresenta uma abordagem para previsão da ocorrência de precipitação na cidade de Manaus baseada na utilização de RNAs. Optou-se por adotar o modelo de *Redes Neurais Multicamadas Feedforward*, as quais foram treinadas com dados oriundos de cerca de 40 anos de registros climatológicos, que incluem temperatura máxima, mínima, umidade, etc. Foram construídas 2500 RNAs com arquiteturas diferentes e o objetivo era encontrar a rede com maior porcentagem de acertos na previsão de chuva. Como resultado, a melhor rede encontrada neste trabalho possui arquitetura 4 – 9 – 7 – 1 e é capaz de prever precipitações com cerca de 70% de precisão.

Para apresentar estes resultados, o trabalho está organizado como segue. Uma breve fundamentação sobre RNAs é mostrada na Seção 2. A Metodologia adotada neste trabalho encontra-se descrita na Seção 3.3. Os resultados são apresentados e discutidos na Seção 4. Por fim, as considerações finais e trabalhos futuros encontram-se na Seção 5.

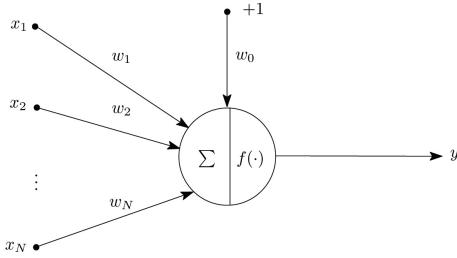
## 2. Redes Neurais Artificiais

Inspiradas no cérebro humano, no qual neurônios realizam contínuas computações paralelas, as *Redes Neurais Artificiais* (RNAs) são sistemas paralelos e distribuídos compostos por unidades de processamento estruturalmente simples e que computam determinadas funções matemáticas.

Para resolver um problema utilizando este modelo de computação, inicialmente são apresentados exemplos à rede. Durante esta fase, chamada *fase de treinamento*, a rede extrai características e padrões relevantes sobre a entrada. Posteriormente, durante as *fases de validação e testes*, estas informações serão utilizadas para obter respostas sobre o problema proposto [de Pádua Braga et al. 2007].

Um *neurônio artificial* é a unidade fundamental de processamento de informação em uma rede neural. O modelo de neurônio mais simples provê uma combinação linear de  $N$  pesos  $w_1, \dots, w_N$  e  $N$  entradas  $x_1, \dots, x_N$ , cujo resultado é passado por uma função não-linear  $f$ , como mostrado na Figura 1.

A entrada de um neurônio é um vetor  $\mathbf{x} = [x_1, \dots, x_N, 1]^T$ , enquanto  $\mathbf{w} = [w_1, \dots, w_N, w_0]^T$  é um vetor de pesos do neurônio. O peso  $w_0$  é o peso que corresponde ao viés da entrada, normalmente sendo igual a 1. O operador de soma realiza um mapeamento  $\mathbb{R}^n \rightarrow \mathbb{R}$ . A função  $f : \mathbb{R} \rightarrow (0, 1)$ , chamada *função de ativação*, é monótona,



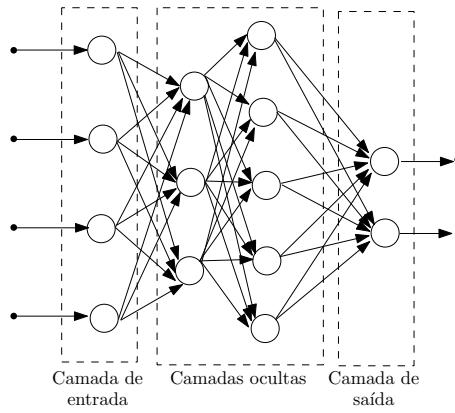
**Figura 1: Representação de um neurônio artificial.**

contínua e sigmóide, na maioria dos casos [Mandic and Chambers 2001]. A saída  $y$  é obtida por meio da seguinte equação:

$$y = f \left( \sum_{i=1}^N w_i \cdot x_i + w_0 \right). \quad (1)$$

Na computação neural, este tipo de neurônio segue o modelo proposto por McCulloch e Pitts [McCulloch and Pitts 1943]. Neurônios individuais, porém, possuem capacidade computacional limitada. Entretanto, quando conectados em uma rede – *rede neural* – são capazes de resolver problemas de mais alta complexidade [de Pádua Braga et al. 2007]. Modelos de redes neurais são especificados pela topologia da rede, características dos neurônios e regras de aprendizagem ou treinamento [Mandic and Chambers 2001].

Um dos modelos de redes neurais, as chamadas *Redes Neurais Multicamadas Feedforward* (RNMF), são caracterizadas pela presença de uma ou mais *camadas ocultas*, compostas por certa quantidade de neurônios, conforme ilustrado na Figura 2. O termo “oculto” refere-se ao fato de que estas camadas não são vistas diretamente a partir da entrada ou da saída da rede. A função destas camadas ocultas de neurônios é intervir entre a entrada externa e a saída da rede, permitindo a extração de estatísticas de alta-ordem [Haykin 2009].



**Figura 2: Exemplo de uma rede neural com múltiplas camadas e arquitetura 4 – 3 – 5 – 2.**

As RNAs se aplicam a problemas em que existem dados, experimentais ou gerados por meio de modelos, por meio dos quais a rede adaptará os seus pesos visando a execução de uma determinada tarefa. Há cinco grupos principais de tarefas às quais as RNAs se aplicam: classificação, categorização, aproximação, previsão e otimização. Considerando estas tarefas, há aplicações das RNAs em diversos setores e problemas, dentre os quais se

destacam diagnóstico médico, detecção de fraudes, análise de expressão gênica, agrupamento de clientes, previsão financeira, previsão do tempo, dentre outros [de Pádua Braga et al. 2007].

### 3. Predição de Precipitação em Manaus, Amazonas

Manaus é a capital do estado do Amazonas no norte do Brasil e está localizada na região do baixo platô da planície do Rio Negro. Fundada por volta de 1693 hoje é a 7<sup>a</sup> cidade mais populosa do país com cerca de 2 milhões de habitantes no ano de 2014 [Moreira and de Sene 2010].

De acordo com o sistema de classificação climática de *Köppen*, Manaus possui um clima tropical de monções, devido às altas temperaturas constantes durante todo o ano e as intensas chuvas de outubro a junho. Seu clima não categoriza-se como de floresta tropical devido ao mês mais seco do ano, agosto, o qual apresenta menos de 60 mm de precipitação [Alvares et al. 2014].

A região da cidade de Manaus e seus arredores possuem condições climáticas naturalmente desagradáveis, com calor e umidade característicos de regiões equatoriais, sendo ocasionados principalmente por dois fatores: sua localização geográfica e topografia [da Silva 2012]. Muitas de suas precipitações são resultados de vários sistemas de precipitação, como a zona de convergência Intertropical, Anticíclopes Subtropicais, Alta da Bolívia, Sistemas de Mesoescala Tropical, Zona de Convergência do Atlântico Sul, Sistemas de Escala Sinóptica, dentre outros [da Silva 2012]. A chuva na cidade de Manaus é abundante, não uniforme e um importante fator que modela o clima da região [Silioli 1991], porém, constata-se que é um processo resultante de sistema natural de elevada complexidade.

#### 3.1. Dados de Entrada e Saída

Os dados utilizados para a realização deste trabalho provêm de uma *estação meteorológica automática* composta pelos subsistemas de: (i) coleta de dados, cujo objetivo é integrar diversos sensores que aferem os parâmetros climáticos de interesse; (ii) armazenamento, que mantém os dados em uma memória não volátil; (iii) comunicação, cuja tarefa é a transmissão dos dados da estação; e (iv) energia, responsável por manter a estação funcionando por meio de energia solar. Os dados produzidos pela estação são validados e então armazenados em um banco de dados disponibilizado pelo INMET (Instituto Nacional de Meteorologia) [INMET 2015a].



**Figura 3: Estação Meteorológica Automática utilizada pelo INMET.**

Considerando o problema da previsão de precipitação na cidade de Manaus, foi utilizada a estação meteorológica do INMET, localizada em 3°6'13.2552"S, 60°0'6596"W, a 61.25m acima do nível do mar [INMET 2015b]. Esta estação encontra-se ilustrada na Figura 3. Foram recuperados dados climáticos registrados diariamente entre 01 de Janeiro

de 1970 a 31 de Dezembro de 2014, resultando num conjunto com cerca de 13179 dias de dados válidos. A partir destas observações, foram colhidos os seguintes parâmetros meteorológicos:

1. Temperatura Máxima Média;
2. Temperatura Mínima Média;
3. Umidade Relativa Média;
4. Velocidade Média do Vento;
5. Precipitação Média.

No contexto deste trabalho, os parâmetros de 1 a 4 serão utilizados como entradas das redes neurais, enquanto que o parâmetro 5 será a saída, ou seja, o valor que se deseja prever.

### 3.2. Modelo de Redes Neurais

O modelo de *Redes Neurais Multicamadas Feedforward* (RNMF) será utilizado para realizar a tarefa de predição proposta. Este modelo foi escolhido por ser capaz de aproximar, com precisão aceitável, uma função arbitrária [Barreto 2002].

A utilização do modelo RNMF em tarefas de previsão de chuva já mostra-se consolidada na literatura. Trabalhos como o de Abhishek et al. [Abhishek et al. 2012] e Luk et al. [Luk et al. 2001] mostram que RNMFs apresentam resultados similares quando comparadas a modelos de maior complexidade em problemas de predição de chuva. Estes resultados reforçam a escolha do modelo em questão no contexto deste trabalho.

No trabalho de Luk et al. [Luk et al. 2001], em particular, este modelo de redes neurais foi capaz de gerar alertas contra inundações com 15 minutos de antecedência. Já no trabalho de Abhishek et al. [Abhishek et al. 2012], estas redes foram utilizadas para prever monções na Índia. Ambos os trabalhos mostram aplicações práticas e relevantes das RNMFs.

### 3.3. Metodologia

Antes de apresentar os dados para o treinamento das redes neurais, um processo de *normalização pela média* foi realizado, conforme

$$x'_i = \frac{x_i - \mu}{\sigma}, \quad \forall x_i \in X, \quad (2)$$

em que  $X$  é o conjunto de dados a ser normalizado,  $\mu$  e  $\sigma$  são os valores da média aritmética e do desvio padrão do conjunto  $X$ , respectivamente. Isto ocorreu para reduzir a grande variabilidade dos dados, visando a melhor convergência do algoritmo de retropropagação de erros (*backpropagation*).

Os 13179 dias de dados válidos foram separados em três conjuntos: (i) *treinamento*, cujos dados apresentarão as características do problema à rede; (ii) *validação*, cujos dados servirão para avaliar o aprendizado da rede; e (iii) *teste* do qual os dados são destinados a averiguar a capacidade de generalização da rede. A separação destes conjuntos visou segmentar de maneira temporal o conjunto de dados de entrada. Desta forma, o conjunto de treinamento recebeu dados entre 1970 e 2010 (86% dos dados); ao conjunto validação foram assinalados os anos de 2011 a 2013 (cerca de 11%); enquanto que ao conjunto teste foram designados 365 dias de dados (aproximadamente 3%), relativos ao ano de 2014. Tal separação foi feita com o objetivo de agrupar características de memória inerentes aos

processos climáticos, ao mesmo tempo que tenta-se evitar o *overfitting*, uma vez que cada dado do conjunto será apresentado à rede de forma randômica.

Embora redes neurais com somente uma camada escondida possam aproximar qualquer função contínua, optou-se por utilizar redes com duas camadas escondidas na expectativa de que estas pudessem capturar melhor os detalhes envolvidos no mapeamento de uma função de predição. Para isso, foram treinadas redes com  $X = 4$  neurônios na camada de entrada,  $Y$  e  $Z$  neurônios na primeira e segunda camadas escondidas, respectivamente, em que  $1 \leq Y, Z \leq 50$ . Há apenas 1 neurônio na camada de saída, correspondendo a previsão realizada pela rede.

Um total de 2500 redes neurais diferentes poderiam satisfazer os requisitos previamente apresentados. Para automatizar a tarefa de treinamento, validação e testes dessas redes, bem como para a posterior análise dos dados, o software MATLAB® foi utilizado.

Dado o amplo número de redes neurais a considerar, a escolha das arquiteturas mais apropriadas foi efetuada em duas etapas:

1. **Treinamento.** Considerou o *Erro Médio Quadrático* (MSE - *Mean Squared Error*) da previsão de chuva como parâmetro guia para a avaliação do aprendizado;
2. **Avaliação.** Nesta etapa, as redes foram analisadas tendo como medida de performance a porcentagem de acertos na fase de testes.

O MSE não poderá ser usado como métrica confiável para a escolha das redes, pois, por tratar-se de um trabalho de classificação, suas saídas fornecerão a predição da probabilidade de ocorrer precipitação em um determinado dia. Mesmo em problemas de regressão um baixo MSE pode ser uma medida de performance errônea, principalmente nos casos onde ocorre *overfitting*. Por este motivo, as arquiteturas de rede selecionadas serão as que possuírem maior porcentagem de acertos na fase de testes e que, por conseguinte, efetuaram a tarefa de predição mais precisamente.

As redes foram treinadas utilizando a otimização de *Levenberg-Marquardt* que é uma aproximação do método de Newton e que utiliza uma taxa de aprendizagem variável [Fun and Hagan 1996]. Geralmente é o algoritmo *backpropagation* mais rápido, embora requeira mais memória que outros algoritmos disponíveis [Mathworks 2015]. Utilizou-se a função tangente sigmoidal como função de ativação dos neurônios da rede, sendo esta muito utilizada em problemas de classificação como o abordado neste trabalho [Darji et al. 2015].

Os parâmetros de treinamento da rede foram mantidos fixos e encontram-se listados na Tabela 1, sendo a diferenciação entre as diversas redes testadas dada pela variação do número de neurônios das camadas escondidas.

Utilizando a metodologia proposta, dentre as 2500 redes neurais consideradas, aquelas com melhor performance para previsão de chuva em Manaus foram identificadas. Os resultados obtidos, suas descrições e análises são apresentadas na seção a seguir.

#### 4. Resultados e Discussões

Foi realizada uma busca exaustiva no conjunto das possíveis arquiteturas de redes neurais a serem utilizadas como um modelo de predição. Uma vez que o treinamento fora concluído, os resultados produzidos foram avaliados resultando na seleção de 3 arquiteturas com melhor desempenho, no contexto das métricas de avaliação adotadas, cujas informações encontram-se na Tabela 2.

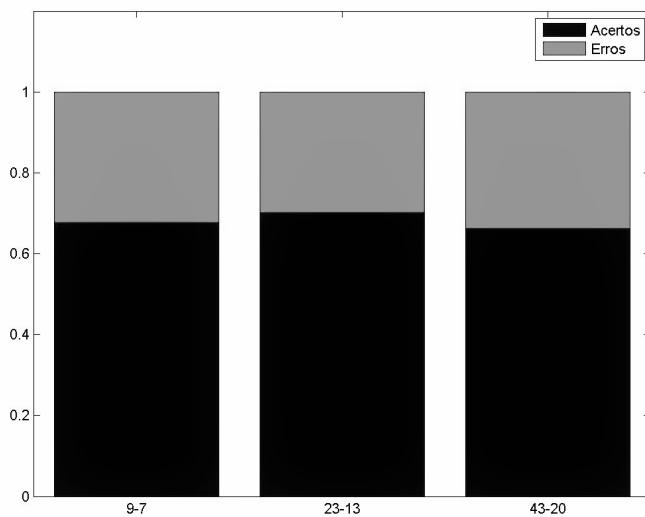
**Tabela 1: Parâmetros utilizados no treinamento das redes neurais.**

Parâmetro	Valor
erro desejado	$0.5 \times 10^{-4}$
épocas de treinamento	1000
erros máximos	7
gradiente mínimo	$1 \times 10^{-10}$
$\mu$ (taxa de aprendizado)	0.003
decremento do $\mu$	0.01
incremento do $\mu$	7
$\mu$ máximo	$1 \times 10^{10}$
tempo de treinamento	indeterminado

**Tabela 2: Arquiteturas de redes selecionadas e seus parâmetros de avaliação**

Arquitetura	MSE	Acertos (%)
9 – 7	0.1930	67.67
23 – 13	0.1870	70.14
43 – 20	0.2069	66.30

A porcentagem de acertos, apesar de se mostrar levemente semelhante entre as diversas arquiteturas verificadas, possui maior variabilidade, com desvio padrão de 0.0367, fornecendo um melhor *insight* a respeito do desempenho da arquitetura da rede. A comparação de performance das arquiteturas selecionadas baseadas neste parâmetro encontra-se na Figura 4. Nela, podemos constatar que mesmo um aumento substancial de neurônios nas duas camadas escondidas pode não produzir melhora significativa na predição das redes, como no caso da terceira arquitetura.

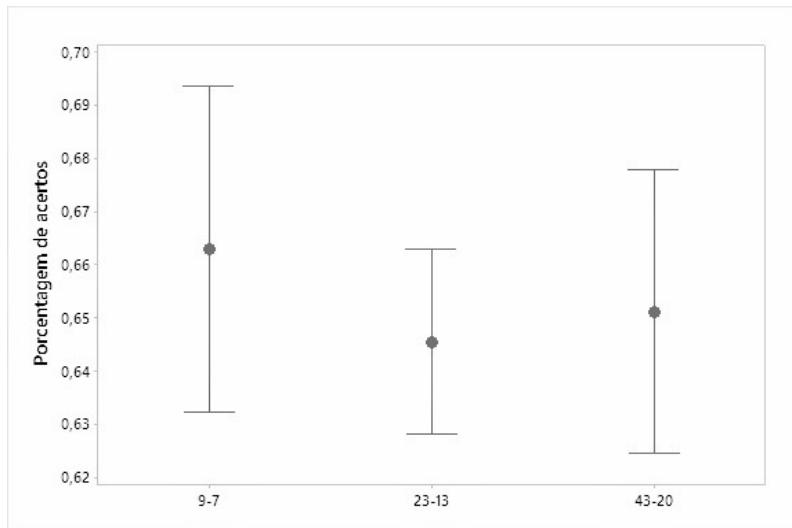


**Figura 4: Comparação entre do desempenho das três redes selecionadas.**

A fim de definir a melhor arquitetura, dentro dos parâmetros avaliados neste trabalho, seria necessário estimar, de forma mais confiável, os resultados produzidos pelas

redes neurais selecionadas. Para isso, foram gerados os intervalos de confiança de cada arquitetura, que executaram 10 previsões correspondentes aos dias do ano 2014, das quais foram salvas as porcentagens de acerto, como amostras. Uma vez que o nível de significância empregado foi de  $\alpha = 0.05$ , pode-se afirmar que fora obtido, de forma fundamentada, limites de confiança que conterão o valor de acerto médio das redes neurais com uma certeza de 95%.

Analizando a Figura 5, na qual os intervalos de confiança das três redes encontram-se denotados, nota-se que as médias inferidas dos percentuais de acerto de cada arquitetura de rede podem englobar-se dentro dos intervalos de confiança umas das outras, o que demonstra certa equivalência entre o desempenho das redes. Por esta razão, é possível afirmar que a arquitetura de rede neural mais adequada para a tarefa de predição proposta é a que contém 9 e 7 neurônios em sua primeira e segunda camada intermediária, respectivamente. Esta rede foi escolhida por possuir menor número de neurônios que as outras redes, o que ocasiona menor esforço computacional, enquanto apresenta considerável capacidade de generalização.

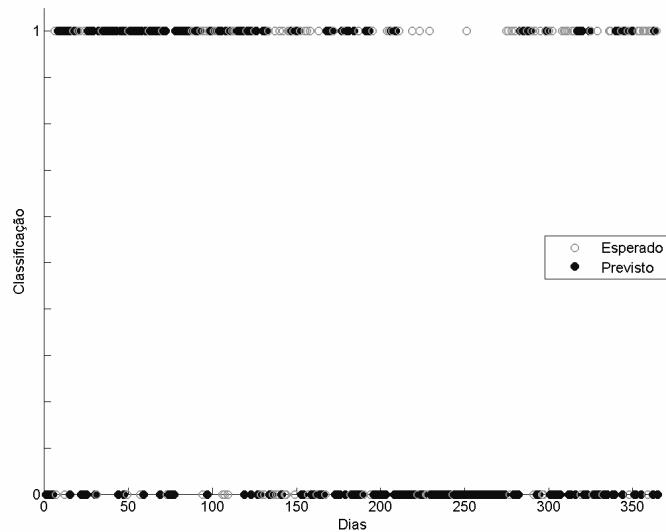


**Figura 5: Intervalos de confiança para as arquiteturas de rede selecionadas. No eixo X, a indicação da arquitetura da rede; no eixo Y o percentual de acertos da rede.**

A rede apresentou boa convergência em seu treinamento não tendo muitas flutuações. Uma demonstração da previsão de chuva é dada na Figura 6, onde pode-se observar a relação entre os dados de precipitação observados no ano de 2014 e os dados previstos pela rede identificada no escopo deste trabalho.

## 5. Considerações Finais

Neste trabalho foi ilustrada a aplicação de redes neurais multicamadas *feedforward* ao problema de previsão de precipitações na cidade de Manaus, Amazonas. Para endereçar este problema, foram utilizados dados obtidos de uma estação meteorológica localizada na cidade em questão durante os anos de 1970 a 2014. Como entrada, considerou-se os dados diários de temperatura máxima, temperatura mínima, umidade relativa e velocidade do vento. Como saída, considerou-se a ocorrência ou não de precipitação no dia em questão. Os dados foram separados em três conjuntos, sendo o ano de 2014 utilizado para os testes dos dados.



**Figura 6: Resultados observados versus previsto da rede 4 – 9 – 7 – 1 para o ano de 2014.**  
**No eixo X os dias observados, enquanto que no eixo Y, temos a classificação dos dias de dados, onde  $Y = 1$  indica presença de precipitação e  $Y = 0$  ausência.**

Considerando este cenário, foram consideradas arquiteturas de redes neurais com 2 camadas escondidas e com até 50 nós em cada uma dessas camadas, resultando em um total de 2500 redes. Os dados foram apresentados à estas redes e três delas foram classificadas como tendo melhor desempenho.

Dentre as três redes identificadas, os intervalos de confiança para o número de acertos das mesmas foi construído e observou-se que, ao nível de significância de 95%, elas são equivalentes em termos de desempenho. Assim, selecionou-se a rede com arquitetura 4 – 9 – 7 – 1 como sendo a melhor dentre as três por possuir menos neurônios. Os resultados obtidos neste trabalho mostraram que, com a utilização de redes neurais, foi possível prever chuvas em Manaus com até 70% de acerto.

Em trabalhos futuros, almeja-se aumentar a porcentagem de acertos das redes. Para tanto, espera-se considerar a recorrência e a relação temporal entre os dados de entrada. Se possível e disponível, deseja-se considerar mais variáveis climatológicas como entrada. Além disso, a previsão antecipada de chuva em Manaus é algo a ser investigado, considerando não só as redes neurais, como também outros modelos que possam endereçar este tipo de situação.

## Agradecimentos

O autor Patrick Magalhães agradece o apoio financeiro provido pela Samsung para a realização deste trabalho. Os autores agradecem a Profa. Maria Betânia Leal pelas sugestões fornecidas e que contribuíram para a melhoria deste trabalho.

## Referências

- Abhishek, K., M.P.Singh, Ghosh, S., and Anand, A. (2012). Weather forecasting model using artificial neural network. *Procedia Technology*, 4:311–318.

- Alvares, C. A., Stape, J. L., Sentelhas, P. C., Gonçalves, J. L. M., and Sparovek, G. (2014). Koppen's climate classification map for brazil. *Meteorologische Zeitschrift*, 22(6):711–728.
- Barreto, J. M. (2002). Introdução às redes neurais artificiais. UFSC - Universidade Federal de Santa Catarina.
- da Silva, D. A. (2012). Função da precipitação no conforto do clima urbano da cidade de manaus. *Revista Geonorte*, 1(5):22–40.
- Darji, M. P., Dabhi, V. K., and Prajapati, H. B. (2015). Rainfall forecasting using neural network: A survey. In *International Conference on Advances in Computer Engineering and Applications*, India. IMS Engineering College.
- de Pádua Braga, A., de Leon F. de Carvalho, A. P., and Ludermir, T. B. (2007). *Redes Neurais Artificiais: Teoria e Aplicações*. LTC, 2 edition.
- Fun, M. H. and Hagan, M. T. (1996). Levenberg-marquardt training for modular networks. *IEEE International Conference on Neural Networks*, 1:468–473.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. 3 edition.
- INMET (2015a). Instituto nacional de meteorologia. [http://www.inmet.gov.br/portal/css/content/topo\\_iframe/pdf/Nota\\_Tecnica-Rede\\_estacoes\\_INMET.pdf](http://www.inmet.gov.br/portal/css/content/topo_iframe/pdf/Nota_Tecnica-Rede_estacoes_INMET.pdf).
- INMET (2015b). Instituto nacional de meteorologia. <http://www.inmet.gov.br>.
- Luk, K. C., Ball, J. E., and Sharma, A. (2001). An application of artificial neural networks for rainfall forecasting. *Mathematical and Computing Modelling*, 33:683–693.
- Mandic, D. P. and Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction*. Wiley.
- Mathworks (2015). Levenberg-marquardt backpropagation. <http://www.mathworks.com/help/nnet/ref/trainlm.html>.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133.
- Moreira, J. C. and de Sene, E. (2010). *Geografia Geral e do Brasil*. Editora Scipione.
- Sioli, H. (1991). *Amazônia: Fundamentos da ecologia da maior região de florestas tropicais*. Editora Vozes.

# Neural Networks for Time Series Rainfall Forecasting: A Case Study in Manaus, Amazonas

Elloá B. Guedes, Patrick Magalhães de Lima, Maria Betânia Leal de Oliveira

<sup>1</sup>Superior School of Technology  
Amazonas State University  
Av. Darcy Vargas, 1200 – Manaus – Amazonas

{ebgcosta, pml.eng, mloliveira}@uea.edu.br

**Abstract.** In this paper we aim at forecasting rainfall occurrence using time series and neural networks. In our approach, three meteorological variables (average high and low temperatures and relative humidity) are taken as input by neural networks with a single hidden layer in order to deliver one-step ahead rainfall occurrence predictions. We performed a case study to evaluate such approach considering 40 years of data from an automated weather station located in Manaus, Amazonas. From 38 neural networks suited to this scenario, we could identify, using Akaike's Information Criterion, that the one with window size equal to 3 and 7 neurons in the hidden layer could forecast rainfall with an accuracy of 99.71% for a testing set. The results obtained indicate the feasibility and efficiency of time series neural networks for rainfall forecasting, suggesting a methodology that can be adopted by many other locations.

## Introduction

Forecasting rainfall is a very important task. It is a crucial activity for preventing flooding and managing water resources. The variations in timing and quantity of rainfall have potential impact on agricultural activities [Darji et al. 2015]. Given such importance, mankind has attempted to predict the weather since ancient times. However, forecasting meteorological events can be a very complex activity because of atmospheric instability, nonlinear interactions between different spatial scales, and even human interference [Bushara and Abraham 2013].

Although the complexity involved, forecasting rainfall has been extensively explored by different approaches. Statistical methods, for instance, consider the analysis of the rainfall distribution according to a time series and its characteristics, such as seasonality, linearity, trend, and stationarity [Palit and Popovic 2005]. Computational Intelligence has also been used to address such problem in many places around the world. Neural networks, in particular, have caught attention for being good forecasters because they are data driven, capable to generalize, computationally efficient and can detect nonlinear relationship among input and output variables [Darji et al. 2015].

Considering the role of Computational Intelligence to rainfall forecasting, in this paper we use artificial neural networks and time series to deal with the problem of rainfall occurrence in a case study scenario. We considered data from Manaus, Amazonas, Brazil that has a tropical monsoon climate in which rainfall is very abundant, but not uniform. We used past lagged observations from 40 years of three atmospheric variables to train 38 different neural networks. According to our results, it was possible to see that the most adequate neural network for such scenario, according to Akaike's Information Criterion, has 7 neurons in the hidden layer and considers data from 3 previous days, delivering

results for a testing set with accuracy of 99.71%. The results obtained indicate the feasibility and efficiency of time series neural networks for rainfall forecasting, suggesting a methodology that can be adopted by many other locations.

To present such results, this paper is organized as follows. An overview of time series in the context of neural networks is introduced in Sect. 2. Some works in the literature that consider time series and neural network for rainfall forecasting are discussed in Sect. 3. A detailed characterization of our problem, including domain and input data, as well as the methodology adopted in order to obtain the results are shown in Sect. 4. Results obtained and the discussion are presented in Sect. 5. Lastly, final remarks and suggestion for future work can be found in Sect. 6.

## Neural Networks for Time Series Forecasting

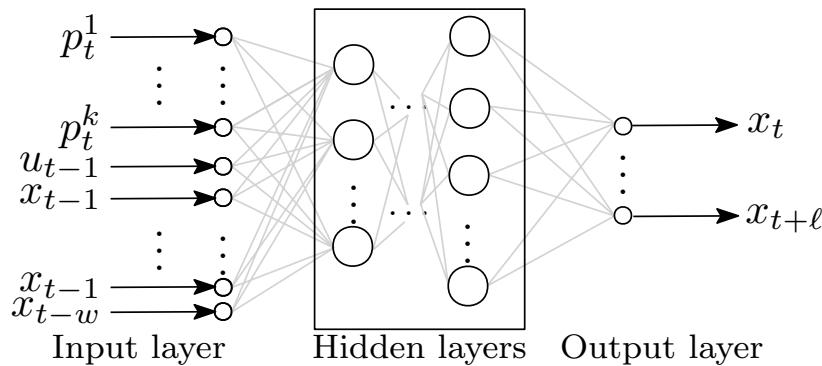
Neural networks have been widely used as time series forecasters. In 1964, Hu [Hu 1964] was the first to demonstrate – on a practical weather forecasting example – the general forecasting capability of neural networks. Since so, many successful applications on market predictions, meteorological and network traffic forecasting have been described in literature [Frank et al. 2001].

A time series is a time-ordered sequence of observation values of a physical or financial variable made at equally spaced time intervals  $\Delta t$ , represented as a set of discrete values  $x_1, x_2, x_3, \dots, x_{t-1}$ . We are interested in forecasting a future value  $x_t$  based on these previous inputs. In a formal approach, we want to find a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  such as to obtain  $x_t$  from the  $n$  time steps back from time  $t$ :

$$x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-w}) \quad (1)$$

The classical approach for time series predictions considers statistical methods, such as the Box-Jenkins that is a general forecasting methodology for time series generated by a stationary autoregressive moving-average process [Palit and Popovic 2005]. However, statistical approaches lack an ability to identify nonlinear patterns and irregularity in time series [Darji et al. 2015]. Several works in the literature verified that neural networks have ability to implicitly detect complex nonlinear relationships between dependent and independent variables. As a result, when compared to classical methods, it could be verified that: (i) for time series with long memory, both approaches deliver similar results; (ii) neural networks outperform the classical Box-Jenkins approach in some experiments by more than 100%; and (iii) the optimally tuned neural network topologies are of higher efficiency than the corresponding traditional algorithms [Palit and Popovic 2005].

A neural network for time series forecasting aims at inducing a function  $f$  with the same idea as shown in Eq. (1), but with some particularities. In order to forecast a certain  $x_t$ ,  $k$  parameters  $p_t^1, p_t^2, \dots, p_t^k$  collected at time  $t$  are given together with some previous inputs  $u_{t-1}, u_{t-2}, \dots, u_{t-w}$  and their respective outputs  $x_{t-1}, x_{t-2}, \dots, x_{t-w}$  of the phenomena being predicted. If the focus is on *one-step-ahead forecasting*, then only one output neuron is needed, which will return  $x_t$ ; otherwise, for *multi-step-ahead forecasting*,  $\ell + 1$  neurons may be employed in the output layer, returning  $x_t, x_{t+1}, \dots, x_{t+\ell}$ . This strategy to use neural network with time series is shown in Figure 1. The number  $w$  of past lagged input and output observations taken by the neural network will be called *window size* from now on.



**Figure 1: Architecture of a multilayer feedforward neural network to forecast data from a time series. The number of hidden layers and their neurons can vary according to the scenario considered.**

Determining the neural network architecture in a time series forecasting scenario is a challenging task. One must decide: (1) what kind of forecasting will be considered (one-step or multi-step ahead) because it will define the number of neurons in the output layer; (2) how many and what past lagged observation will be taken as input (*window size estimation*); (3) the number of hidden layers and of neurons in each layer – these parameters are useful for capturing the nonlinear relationship between input and output variables; and (4) the type of neural network (Multi-Layer Perceptron, Radial Basis Function, etc.).

Some answers to these questions can be found in the literature. In the scenario of time series forecasting, feedforward neural networks are mostly used. Regarding the number of hidden layers, most forecasting applications use only one hidden layer and a small number of hidden nodes. Perhaps the most difficult part is determining the input data and the window size. Empirical results have been suggesting that the input layer is more important than the hidden layer in time series forecasting problems. After determining the neural network and its architecture, it must be trained so that the parameters of the network can be estimated from the data. The Levenberg-Marquardt optimization for back-propagation is usually a frequent choice due to its good performance, even requiring more computer memory usage [Zhang 2012].

Taking into account the practices described for a time series forecasting scenario, once the neural network has been trained, it can be used for prediction. However, an important question that needs to be discussed is the *performance evaluation* which aims at quantifying the goodness of the neural network considered. Literature suggests to divide the dataset into training examples and testing examples. The training examples will be used to figure out the window size estimation, to adjust the weights, and even to validate the neural network, among others. The testing examples are used to estimate the generalization capability of the neural network. The output of the neural network for the testing examples will be compared with the original outcome presented in the dataset. Some metrics such as *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE) are derived in order to quantify the performance of such neural network, enabling comparisons with other works, models and approaches. However, although widely used, a disadvantage of these metrics is that they rely on a specific testing scenario. In order to overcome such limitation, *Akaike's Information Criterion* (AIC) is frequently considered.

Founded on the concept of entropy from Information Theory, AIC offers a relative measure of the information lost when a given model is used to describe reality and can be said to describe the tradeoff between bias and variance in model construction. It is a basis of comparison and selection among several statistical models where the one having the lowest AIC being considered the best [Panchal et al. 2010]. In a time series forecasting scenario with neural networks, AIC will be used as a measure of the relative quality of each neural network for the dataset being used. Given a collection of different neural networks, AIC will be used to estimate the quality of each model, relative to each of the other models.

Considering the concepts and methodologies introduced for time series forecasting with neural networks, some applications described in the literature considering the problem of rainfall forecasting will be detailed in the next section.

## Related Work

Neural networks have been adopted to rainfall forecasting in many works in the literature, delivering relevant results and being considered a more suitable approach than classical methods commonly used on prediction tasks [Nayak et al. 2013]. In this section we will show some works with the neural network approach to rainfall forecasting.

The work of Abbot and Marohasy considered the monthly and seasonal rainfall forecasting in Queensland, Australia [Abbot and Marohasy 2012]. The authors developed Time-Delay Recurrent Neural Networks (TDRNNs) to forecast rainfall 3-month lead. Their models considered both time-delay inputs and output recurrence to incorporate dynamics and “memory” capability. The input data were divided into four classes: (i) monthly rainfall, corresponding to 20 sites in Queensland State, lagged up to 12 months; (ii) climatic indexes, whose more relevant were Southern Oscillation Index (SOI), Pacific Decadal Oscillation (PDO), Niño 3.4 and Dipole Mode Index; (iii) atmospheric temperature, considering the minimum and maximum temperature values; (iv) solar data, in which sunspot and total solar irradiation were chosen.

The authors developed a TDRNN with two hidden layers using a trial and error approach to evaluate different networks configurations. The resulting model was used to forecast rainfall on each of 20 sites with Normalized Mean Squared Error (NMSE) and Pearson correlation coefficient as performance evaluators. Initially only the monthly rainfall values were used as inputs, other data were progressively added. The best data input combination consisted of lagged monthly rainfall, atmospheric temperature, SOI, PDO and Niño 3.4. The results of TDRNN performance were then compared with the Predictive Ocean Atmosphere Model for Australia (POAMA) and with the General Circulation Model (GCM). It could be observed that the TDRNN model achieved lower NMSE values on 17 sites of a total of 18 sites used in this comparison.

Htike and Khalifa worked in rainfall forecasting models to Malaysia using Time-Delay Neural Networks (TDNN) to perform one-step-ahead predictions [Htike and Khalifa 2010]. The data set, consisting in daily rainfall data from January of 1980 to May of 2009, were obtained from Malaysian Meteorological Department and treated to create four data sets composed by monthly, quarterly, biannual and yearly rainfall data, being their values normalized within the range from -1 to 1 afterward.

In this work, TDNN models with one hidden layer were developed to each data set and the approximated optimal number of input delays and hidden neurons for these models were obtained from systematic trial and error. The authors used 80% of the data

set to train and validate the models and the remaining 20% to test them, also using the MAPE value obtained in this dataset to evaluate the models' accuracy. The yearly forecast model gave the most accurate result (94.25%), which decreased for the biannual, quarterly and monthly models, respectively.

Neural networks were used by Shukla et al. in order to forecast the Indian Summer Monsoon Rainfall Index (ISMRI) using Sea Surface Temperature (SST) anomalies indices, specifically the El Niño-Southern Oscillation (ENSO) [Shukla et al. 2011]. The authors used the ISMRI index corresponding to the period of 1951 to 2003 as well as the Niño-1 + 2, Niño-3, Niño-3.4 and Niño-4 indexes from 1950 to 2003 as input parameters. Principal Component Analysis technique was used to reduce the number of input parameters due to the amount of data considered. This technique generates a vector of uncorrelated values which are called Principal Components (PC), in which the first one – most relevant – was chosen to be used as input of the neural network. All the data were also normalized.

Multilayer feed-forward neural networks with a single hidden layer were used by the authors. In addition, different regression models were constructed, where five of these models used the Niño indexes individually, one used all the input parameters and the last model used only the PC value. These models were compared according to performance metrics such as root mean squared error, correlation coefficient and standard deviation. Although the multiple regression model being the best among this class, its prediction was inferior to the mean prediction. However, the neural network model gave significant results, better than the mean prediction, which reinforced the idea of non-linear relationship between ISMIR and Niño indexes.

The literature surveyed regarding rainfall forecasting with neural networks shows that time-delay neural networks are mostly adopted, with satisfactory accuracy results. In practical scenarios, forecasting rainfall may vary considering the time series data available in the location of interest. Instead of relying in complex meteorological variables with non-trivial measurement and significance, our case study considers attributes that are easily available in many locations, collected from typical automated weather stations.

## Case Study

In this case study, we considered the one-step ahead rainfall forecasting in Manaus, Amazonas. The dataset used was obtained from an automatic weather station hosted and maintained by the National Institute of Meteorology (*Instituto Nacional de Meteorologia – INMET*) located at 3°6'13.2552"S, 60°0'6596"W, 61.25m above sea level, between the years of 1970 to 2010 [INMET 2015]. Such weather station is shown in Figure 2. Missing values in data were discarded, and 11719 entries characterize the dataset considered in this work.

Our goal is to one-step ahead predict the occurrence or not of rainfall in Manaus. This is a very challenging task because the rainfall in Manaus is very abundant and not uniform [Sioli 1991]. Most part of the rainfalls in Manaus are a result of many precipitating systems, such as the Intertropical convergence zone, Subtropical Anticyclones, Bolivian High, Tropical Mesoscale Systems, South Atlantic Convergence Zone, Synoptic Scale Systems, among others [da Silva 2012]. We can see that the rainfall in Manaus is the result of a very complex natural system.

According to the Köppen climate classification system, Manaus has a tropical monsoon climate, with consistently high temperatures throughout the year and intense rainfall



**Figure 2: Automated weather station in Manaus, Amazonas.**

from October through to June. August is the driest month with less than 60mm of precipitation [Alavares et al. 2014]. The rainfall in Manaus is very important factor that shapes the weather in the city [Sioli 1991]. It happens because the city of Manaus and its surroundings have a naturally uncomfortable weather, with warm and humid Equatorial characteristics [da Silva 2012].

## Data Overview

In our case study, we consider the following five attributes in the dataset:

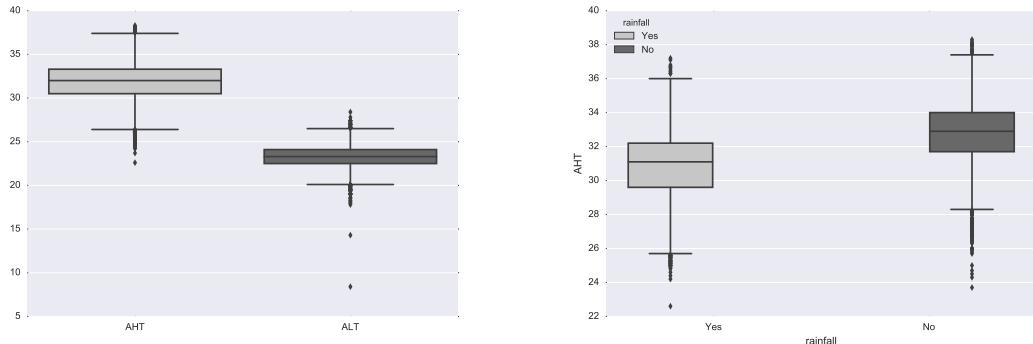
1. Average High Temperature (AHT);
2. Average Low Temperature (ALT);
3. Average Relative Humidity (RH);
4. Average Wind Speed (WS);
5. Rainfall.

The categorical rainfall attribute is the target attribute considered, indicating the occurrence or not of rainfall. The other attributes are numerical and their average and standard deviation are shown in Table 1.

**Table 1: Average and standard deviation of input attributes.**

	AHT	ALT	RH	WS
<b>Average</b>	31.81	23.31	82.97	1.64
<b>Std. Dev.</b>	2.17	1.20	7.84	1.60

The AHT and ALT attributes, which indicate temperatures in the Celsius scale, have their distribution showed in boxplots of Figure 3a. It is possible to notice that there are some overlapping between the values, indicating that a low temperature can be considered as a high depending on seasonal factors. It is also possible to see, in Figure 3b, that there is not a clear separation for the AHT values considering rainfall occurrence. The same was observed for ALT values.



(a) Boxplots of AHT and ALT attributes. (b) Boxplot of the AHT values considering rainfall occurrence.

Figure 3: Boxplots obtained from the dataset considered.

Aiming at verifying which input attributes may have significant relation with rainfall occurrence, we obtained the Pearson correlation coefficient of these attributes versus the rainfall. The results are shown in Table 2. It is possible to notice that there is a significant negative correlation for AHT, ALT and RH attributes. However, the correlation coefficient for WS and rainfall has a negligible value. For that reason, it was decided not to use the attribute WS in the further steps of rainfall prediction.

Table 2: Pearson correlation coefficient of the input attributes versus target attribute.

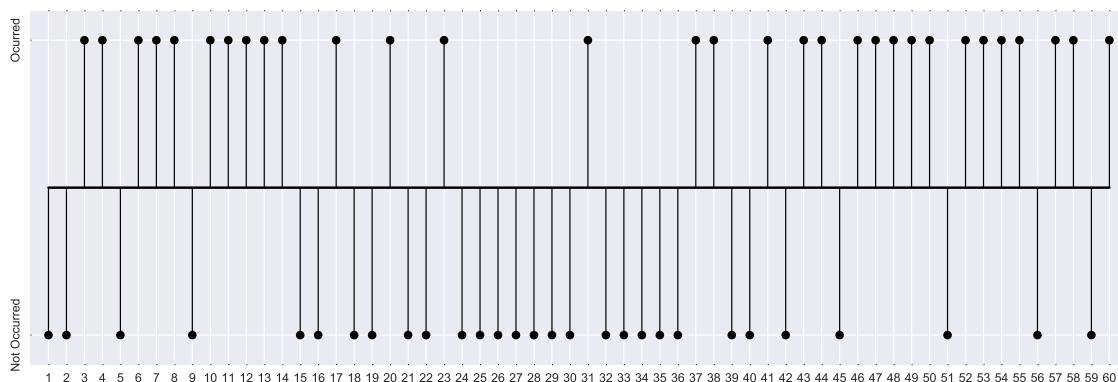
	AHT	ALT	RH	WS
Rainfall	-0.45655336	-0.33254008	-0.33254008	-0.03047564

From 11719 days of observation, it was possible to see that rainfall occurred in 5830 days (49.7%), and not occurred in 5889 days (50.3%). In order to see if it is possible to detect some statistical pattern in the rainfall occurrence, we performed a  $\chi^2$  test, considering as null hypothesis that rainfall is distributed according to a random uniform distribution variable. The  $p$ -value obtained was of 0.585, indicating that the null hypothesis cannot be rejected for a significance level of  $\alpha = 0.05$ . It means that, for a 95% confidence level, it is not possible to distinguish a non-random pattern in rainfall occurrence. In order to illustrate such statistical difficulty in finding a non-random pattern in rainfall, we show a stem plot in Figure 4 of the last 60 days of rainfall in the dataset considered.

## Methodology

In our case study we are going to follow the methodology described below to build, test and train the neural networks for the scenario considered. We are considering the problem of one-step ahead rainfall occurrence forecasting in Manaus.

1. **Window size.** Choose the window size  $n$  from the set [1, 3, 5, 7, 10, 16, 21];
2. **Organize input data.** The main objective of this step is to prepare the input data.
  - (a) Organize the  $n$  past lagged observations as input data: consider the  $n$  previous values of rainfall, AHT, ALT, RH;
  - (b) Add to input data the values of AHT, ALT and RH of the same day that rainfall will be forecasted;



**Figure 4: Stem plot of the rainfall occurrence in the last 60 days of data illustrates the absence of a non-random pattern.**

- (c) Split the entire resulting data set into three subsets considering a temporal division, aiming at provide some temporal features to improve neural network training. The considered range was:
  - i. Training: 1970 - 2006;
  - ii. Validation: 2007 - 2009;
  - iii. Test: 2010.
- 3. **Define several neural network architectures.** Different feedforward neural networks will be considered for this forecasting scenario.
  - (a) **Input Nodes ( $in$ )**. Defined according to the number of input parameters, where:  $in = 4 * n + 3$ , where  $n$  is the window size;
  - (b) **Hidden Layer**. Only a single hidden layer will be considered;
  - (c) **Neurons in the Hidden Layer ( $h$ )**. Defined according to Baum and Haussler rule of thumb [Baum and Haussler 1989] adapted to this scenario:

$$h \leq \frac{N_{TR} \times E_{TOL}}{in + 1} \quad (2)$$

where  $N_{TR}$  is the number of training examples (Step 2-c-i),  $E_{TOL}$  is the error tolerance set to 0.01,  $in$  is the number of input neurons (Step 3-a);

- (d) **Output Nodes**. Set to 1 because one-step-ahead forecasting is being considered.
- 4. **Train each neural network.** Using the Levenberg-Marquardt algorithm and considering the appropriate training set according to the window size defined previously;
- 5. **Validate each neural network.** Discard neural networks whose accuracy may indicate overfitting on the training data;
- 6. **Test each neural network.** Record its performance and hitting percentage on the test set (Step 2-c-iii);
- 7. **Performance Evaluation.** Determine the Akaike's Information Criterion (AIC), MSE and MAPE values for each model.

In the last step, after evaluating AIC, we can rank the neural networks according to it and select the model which has the lower AIC, indicating its best performance for the scenario considered. Despite that, we will also obtain MSE and MAPE because they are considered more intuitive metrics regarding the neural network accuracy, being widely used in this research domain.

## Results and Discussion

By following the methodology previously described, we built and trained 38 neural networks for the scenario of rainfall forecasting in Manaus. The architecture of such networks and their MSE and MAPE results are presented in Table 3, where  $w$  is the window size,  $in$  in the number of neurons in the input layer, and  $h$  is the number of neurons in the hidden layer.

**Table 3: Neural networks results of MSE and MAPE for the testing set.**

$w$	$in$	$h$	MSE	MAPE $\times 10^{-3}$	$w$	$in$	$h$	MSE	MAPE $\times 10^{-3}$
1	7	1	0.1606	2.9911	3	15	7	0.1614	2.8826
1	7	2	0.1570	2.9090	5	23	1	0.1513	2.8893
1	7	3	0.1590	2.8663	5	23	2	0.1497	2.7620
1	7	4	0.1602	3.0250	5	23	3	0.1513	2.8893
1	7	5	0.1613	3.0356	5	23	4	0.1535	2.8135
1	7	6	0.1581	2.9637	5	23	5	0.1512	2.7526
1	7	7	0.1606	2.9600	7	31	1	0.1504	2.8796
1	7	8	0.1673	3.0587	7	31	2	0.1521	2.8591
1	7	9	0.1623	2.8997	7	31	3	0.1496	2.8069
1	7	10	0.1613	2.8997	7	31	4	0.1515	2.8455
1	7	11	0.1622	2.8874	10	43	1	0.1483	2.8711
1	7	12	0.1614	3.0567	10	43	2	0.1492	2.8558
1	7	13	0.1611	3.0332	10	43	3	0.1485	2.8356
3	15	1	0.1540	2.9149	13	55	1	0.1455	2.8531
3	15	2	0.1564	2.9151	13	55	2	0.1425	<b>2.7032</b>
3	15	3	0.1498	2.7936	16	67	1	0.1438	2.8448
3	15	4	0.1552	2.7894	16	67	2	0.1437	2.9010
3	15	5	0.1521	2.8525	21	87	1	<b>0.1422</b>	2.8297
3	15	6	0.1618	2.9722	21	87	2	<b>0.1422</b>	2.8388

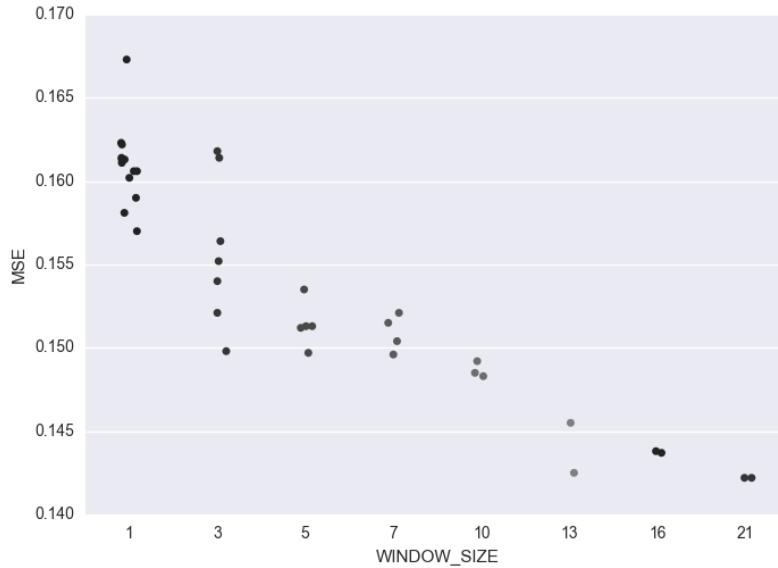
The lower values for MSE in Table 3 can be seen in the neural networks which have the larger window sizes. This result indicate that such larger window size may introduce some *memory* – in a Markovian sense – in the forecasting process, reducing errors. A plot of the window size versus MSE shown in Fig. 5 illustrates such relation observed in the results.

Regarding MAPE, which can be considered as a straightforward method to obtain the accuracy of the models (accuracy = 1 – MAPE), we have that all results are satisfactory on a  $10^{-3}$  order. The best result observed for MAPE was in the neural network with windows size 13 and architecture 55-2-1 whose accuracy for the testing set was of 99.72%. All the MAPE values indicate that the time series approach for rainfall prediction in Manaus is very adequate, delivering correct results rainfall occurrence with a high percentage of correct answers.

## Evaluating Akaike's Information Criterion

Recalling the importance of AIC to identify the best neural network from a set, we need some parameters which are listed below:

1. Number of input nodes ( $in$ );
2. Number of hidden nodes ( $h$ );
3. Number of output neurons ( $o$ );
4. Residual sum of squares ( $RSS$ );



**Figure 5: Window size versus MSE.**

5. Number of data points per training example ( $n$ );

From these parameters we derive  $k$ , where:

$$k = in \cdot h + o \cdot h. \quad (3)$$

If the ratio  $n/k < 40$ , the AIC is obtained as follows:

$$AIC = n \cdot \ln \left( \frac{RSS}{n} \right) + 2 \cdot k + \frac{2 \cdot k \cdot (k+1)}{n - k - 1}, \quad (4)$$

otherwise, AIC is given by:

$$AIC = n \cdot \ln \left( \frac{RSS}{n} \right) + 2 \cdot k. \quad (5)$$

Obtaining the AIC values for the neural networks previously built, we have the results shown in Table 4.

From the lowest AIC, we choose the neural network with window size 3 and architecture 15-7-1 as the most adequate to the rainfall occurrence forecasting in Manaus. Although the results of MSE and MAPE may indicate neural networks with a larger window size than the former, AIC considers a deeper relation between the model and its generalization capacity for the problem under investigation, not only its performance relative to the testing examples. We recall that the accuracy of the neural network 15-7-1 is of 99.71%, only around 0.01% smaller than the best neural network with window size 21. It is important to emphasize some advantages from the result of AIC because it points out a neural network with less neurons than the one pointed by MSE and MAPE; that this neural network was trained with more data (larger  $n$ ); and that it has a smaller window size.

**Table 4: Results of AIC for the neural networks considered.**

<i>w</i>	Architecture	RSS	<i>n</i>	AIC	<i>w</i>	Architecture	RSS	<i>n</i>	AIC
1	7-1-1	1572.70	11718	-23517.63	3	15-7-1	1468.71	11716	<b>-24105.06</b>
1	7-2-1	1548.77	11718	-23681.25	5	23-1-1	1516.32	11714	-23901.20
1	7-3-1	1579.96	11718	-23431.63	5	23-2-1	1496.50	11714	-24007.33
1	7-4-1	1549.33	11718	-23645.02	5	23-3-1	1516.32	11714	-23805.20
1	7-5-1	1537.49	11718	-23718.90	5	23-4-1	1484.46	11714	-24005.93
1	7-6-1	1550.65	11718	-23603.01	5	23-5-1	1494.74	11714	-23877.10
1	7-7-1	1529.21	11718	-23750.20	7	31-1-1	1509.98	11712	-23928.21
1	7-8-1	1569.21	11718	-23431.62	7	31-2-1	1498.47	11712	-23953.78
1	7-9-1	1506.00	11718	-23897.44	7	31-3-1	1480.49	11712	-24031.17
1	7-10-1	1520.02	11718	-23772.80	7	31-4-1	1490.30	11712	-23889.85
1	7-11-1	1523.90	11718	-23726.92	10	43-1-1	1504.87	11709	-23934.75
1	7-12-1	1543.28	11718	-23562.86	10	43-2-1	1497.52	11709	-23904.03
1	7-13-1	1532.91	11718	-23625.87	10	43-3-1	1491.47	11709	-23863.45
3	15-1-1	1530.86	11716	-23811.52	13	55-1-1	1494.82	11706	-23980.03
3	15-2-1	1521.62	11716	-23850.41	13	55-2-1	1472.25	11706	-24046.14
3	15-3-1	1505.62	11716	-23942.28	16	67-1-1	1489.54	11703	-23988.27
3	15-4-1	1497.10	11716	-23976.73	16	67-2-1	1482.00	11703	-23911.69
3	15-5-1	1504.49	11716	-23887.08	21	87-1-1	1480.64	11698	-24003.03
3	15-6-1	1492.98	11716	-23945.01	21	87-2-1	1479.98	11698	-23832.25

## Final Remarks

We addressed the problem of rainfall occurrence forecasting with time series neural networks. We considered the scenario of Manaus from which we used meteorological data from 40 years to train and test 38 different neural networks, considering different window sizes. It was possible to see that the results obtained considering time series delivered a significant accuracy. We were also able to identify that as the window size increases, the MSE for the testing examples decreases. By using Akaike's information criterion, it was possible to choose among the models a 15-7-1 neural network with window size 3 whose accuracy for the problem is equal to 99.71%. The results emphasize and adequation of the considered approach for the problem under consideration.

In future work we aim at using other Computational Intelligence models to deal with the problem under consideration in order to enable comparisons and to search for eventual improvements in the forecasting methodology. We also will use data from weather stations located in other areas of Manaus in order to see how the results can be generalized considering a spatial variation.

## Acknowledgments

The authors acknowledge the National Institute of Meteorology (INMET) for the data used in this work. The authors EBG and MBLO acknowledge the research grant provided by the Amazonas State University that supported this work.

## References

- Abbot, J. and Marohasy, J. (2012). Application of artificial neural networks to rainfall forecasting in Queensland, Australia. In *Advances in Atmospheric Sciences*, volume 29, pages 717–730.
- Alavares, C. A., Stape, J. L., Sentelhas, P. C., Gonçalves, J. L. M., and Sparovek, G. (2014). Köppen's climate classification map for Brazil. *Meteorologische Zeitschrift*, 22(6):711–728.

- Baum, E. B. and Haussler, D. (1989). What size net gives valid generalisation? *Neural Computation*, (1):151–160.
- Bushara, N. O. and Abraham, A. (2013). Computational intelligence in weather forecasting: A review. *Journal of Network and Innovative Computing*, 1:320–331.
- da Silva, D. A. (2012). Função da precipitação no conforto do clima urbano da cidade de Manaus. *Revista Geonorte*, 1(5):22–40.
- Darji, M. P., Dabhi, V. K., and Prajapati, H. B. (2015). Rainfall forecasting using neural network: A survey. In *International Conference on Advances in Computer Engineering and Applications*, India. IMS Engineering College.
- Frank, R. J., Davey, N., and Hunt, S. P. (2001). Time series prediction and neural networks. *Journal of Intelligent and Robotic Systems*, 31:91–103.
- Htike, K. K. and Khalifa, O. O. (2010). Rainfall forecasting models using focused time-delay neural networks. In *2010 International Conference on Computer and Communication Engineering*, pages 1–6. IEEE Press.
- Hu, M. J. C. (1964). Application of the ADALINE system to weather forecasting. Master's thesis, Stanford El. Lab., Stanford, CA.
- INMET (2015). Instituto Nacional de Meteorologia. <http://www.inmet.gov.br>.
- Nayak, D. R., Mahapatra, A., and Mishra, P. (2013). A survey on rainfall prediction using artificial neural network. In *International Journal of Computer Applications*, volume 72, pages 32–40.
- Palit, A. K. and Popovic, D. (2005). *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*. Springer, 1 edition.
- Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D. (2010). Searching most efficient neural network architecture using Akaike's Information Criterion (AIC). *International Journal of Computer Applications*, 1(5):41–44.
- Shukla, R. P., Tripathi, K. C., Pandey, A. C., and Das, I. M. L. (2011). Prediction of Indian summer monsoon rainfall using Niño indices: A neural network approach. In *Atmospheric Research*, volume 102, pages 99–109. Elsevier.
- Sioli, H. (1991). *Amazônia: Fundamentos da ecologia da maior região de florestas tropicais*. Vozes.
- Zhang, G. P. (2012). *Handbook of Natural Computing*, chapter Neural Networks for Time-Series Forecasting. Springer.