



Universidade Federal
do Rio de Janeiro

Escola Politécnica

EXPLORAÇÃO DE ARQUITETURAS DE REDES NEURAIIS EM UMA SÉRIE TEMPORAL FINANCEIRA

Carlos Eduardo Menezes dos Anjos

Projeto de Graduação apresentado ao Curso de Computação e Informação da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

Orientador: Alexandre Gonçalves Evsukoff

Rio de Janeiro
Abril de 2018

EXPLORAÇÃO DE ARQUITETURAS DE REDES NEURAIS EM UMA SÉRIE
TEMPORAL FINANCEIRA

Carlos Eduardo Menezes dos Anjos

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE
COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinadores:

Prof. Alexandre Gonçalves Evsukoff, Dr.

Prof. José Manoel de Seixas, D. Sc.

Prof. Manuel Ramon Vargas, M. Sc.

RIO DE JANEIRO, RJ – BRASIL
ABRIL DE 2018

Menezes dos Anjos, Carlos Eduardo

EXPLORAÇÃO DE ARQUITETURAS DE REDES NEURAIS EM UMA SÉRIE TEMPORAL FINANCEIRA/Carlos Eduardo Menezes dos Anjos. – Rio de Janeiro: UFRJ/POLI – COPPE, 2018.

XI, 60 p.: il.; 29, 7cm.

Orientador: Alexandre Gonçalves Evsukoff

Projeto (graduação) – UFRJ/ Escola Politécnica/ Curso de Computação e Informação, 2018.

Referências Bibliográficas: p. 35 – 39.

1. Redes Neurais.
 2. Aprendizado de Máquina.
 3. Mercado Financeiro.
- I. Gonçalves Evsukoff, Alexandre. II. Universidade Federal do Rio de Janeiro, Escola Politécnica/ Curso de Computação e Informação. III. Título.

Agradecimentos

Agradeço por meio deste a todos que participam e participaram da minha vida e permitiram a criação deste projeto.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

EXPLORAÇÃO DE ARQUITETURAS DE REDES NEURAIIS EM UMA SÉRIE TEMPORAL FINANCEIRA

Carlos Eduardo Menezes dos Anjos

Abril/2018

Orientador: Alexandre Gonçalves Evsukoff

Curso: Engenharia de Computação e Informação

A predição de ações do mercado financeiro é um problema com alto grau de dificuldade devido ao fato da série temporal financeira não ser estacionária e informações externas, como delações vazadas, a afetarem diretamente. Com o avanço da tecnologia ao longo dos anos, verifica-se possível a criação de modelos mais complexos para modelagem desse tipo de séries temporais, de forma que o trabalho aqui proposto visa explorar diferentes topologias e técnicas de redes neurais artificiais em um série temporal financeira brasileira. Os modelos propostos usam o valor de fechamento, junto com alguns indicadores, de cinco dias seguidos, para tentar prever se o valor de fechamento subirá ou descerá no sexto dia. Apesar das técnicas de redes neurais serem consideradas o estado da arte para certos problemas, as redes testadas neste trabalho não apresentaram resultados satisfatórios, visto que apenas a informação apresentada aos modelos não foi suficiente para realizar uma modelagem adequada.

Palavras-Chave: Redes Neurais, Aprendizado de Máquina, Mercado Financeiro.

Abstract of the Undergraduate Project presented to Poli/COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Computer and Information Engineer.

NEURAL NETWORKS ARCHITECTURES EXPLORATION USING STOCK MARKET TIME SERIES

Carlos Eduardo Menezes dos Anjos

April/2018

Advisor: Alexandre Gonçalves Evsukoff

Course: Computer and Information Engineering

The prediction of stock market is a problem with a high degree of difficulty due to the fact that the financial time series is not stationary and external information, such as leaks, affect it directly. With the advancement of technology over the years, it is possible to create more complex models for modeling this type of time series, so that the work proposed here aims to explore different topologies and techniques of artificial neural networks in a Brazilian financial time series. The proposed models use the closing value, along with a few indicators, of five consecutive days to try to predict whether the closing value will rise or fall on the sixth day. Although the neural network techniques are considered the state of the art for certain problems, the networks tested in this work did not present satisfactory results, since only the information presented to the models was not sufficient to carry out an adequate modeling.

Keywords: Neural Networks, Machine Learning, Financial Market.

Sumário

Lista de Figuras	ix
Lista de Tabelas	x
1 Introdução	1
1.1 Motivação	2
1.2 Proposta e Objetivo	2
1.3 Organização do documento	2
2 Revisão Bibliográfica	3
2.1 Redes Neurais Feedforward	3
2.1.1 Perceptron	3
2.1.2 Perceptron de Múltiplas Camadas	5
2.2 Redes Neurais Recorrentes	6
2.2.1 LSTM	7
2.2.2 GRU	8
2.3 Funções Custo	9
2.3.1 Erro Médio Quadrático	9
2.3.2 Entropia Cruzada	9
2.4 Métodos de Atualização de Parâmetros	10
2.4.1 Gradiente Descendente	10
2.4.2 Momentum	11
2.4.3 Adam	11
2.5 Algoritmos de Treinamento	12
2.5.1 Algoritmo de Retropropagação	13
2.5.2 Retropropagação através do Tempo	14
2.6 Normalização de Batelada	15
2.7 Dropout	15
2.8 Trabalhos Relacionados	16

3	Dados e Ferramentas	17
3.1	Ferramentas	17
3.1.1	Python	18
3.1.2	Anaconda	18
3.1.3	Tensorflow	19
3.2	Dados	19
3.2.1	Extração dos Dados	20
3.2.2	Dados da Ação	20
3.2.3	Indicadores	22
4	Análise de Resultados	24
4.1	Procedimento Experimental	24
4.1.1	Modelos	24
4.1.2	Treinamento	26
4.2	Figuras de mérito	27
4.3	Análise de Resultados	29
5	Conclusão	33
5.1	Conclusão	33
5.2	Trabalhos Futuros	33
	Referências Bibliográficas	35
A	Tabelas de Resultados	40
A.1	MLP	40
A.2	GRU	46
A.3	LSTM	52
A.4	Linear	57
B	Modelos aplicados a série solar	58
B.1	MLP	59
B.2	LSTM	59
B.3	GRU	60

Lista de Figuras

2.1	<i>Perceptron ou neurônio simples.</i>	4
2.2	<i>Função sigmoíde.</i>	5
2.3	<i>Exemplo do Perceptron de Múltiplas Camadas.</i>	6
2.4	<i>Arquitetura de uma célula LSTM.</i>	7
2.5	<i>Exemplificação visual do algoritmo de backpropagation.</i>	14
3.1	<i>Exemplo do arquivo de entrada, visualização feita com Pandas.</i>	20
3.2	<i>Representação visual da série temporal dos preços de fechamento.</i>	21
3.3	<i>Representação visual da autocorrelação dos preços de fechamento da ação BBAS3.</i>	21
4.1	<i>Representação visual das topologias escolhidas e usadas.</i>	25
4.2	<i>Representação visual do treinamento aplicado nas redes neurais criadas.</i>	27
4.3	<i>Resultados do melhor Índice SP por quantidade de neurônios na camada escondida.</i>	31
4.4	<i>Resultados do melhor Índice SP por otimizador.</i>	31
4.5	<i>Resultados do melhor Índice SP por taxa de dropout utilizada.</i>	32
4.6	<i>Resultados do melhor Índice SP com e sem normalização.</i>	32
B.1	<i>Representação visual da série temporal solar.</i>	58

Lista de Tabelas

4.1	Matriz de Confusão	27
4.2	Melhores modelos de acordo com o Índice SP.	29
4.3	Melhores modelos LSTM de acordo com o Índice SP.	30
4.4	Melhores modelos GRU de acordo com o Índice SP.	30
4.5	Resultados do modelo Linear no conjunto de teste.	30
A.1	Resultados do modelo MLP no conjunto de teste.	40
A.2	Resultados do modelo MLP no conjunto de treino.	41
A.3	Resultados do modelo MLP com normalização no conjunto de teste.	41
A.4	Resultados do modelo MLP com normalização no conjunto de treino.	41
A.5	Resultados do modelo MLP com normalização e dropout de 60% no conjunto de teste.	42
A.6	Resultados do modelo MLP com normalização e dropout de 60% no conjunto de treino.	42
A.7	Resultados do modelo MLP com normalização e dropout de 30% no conjunto de teste.	43
A.8	Resultados do modelo MLP com normalização e dropout de 30% no conjunto de treino.	43
A.9	Resultados do modelo MLP com dropout de 60% no conjunto de teste.	44
A.10	Resultados do modelo MLP com dropout de 60% no conjunto de treino.	44
A.11	Resultados do modelo MLP com dropout de 30% no conjunto de teste.	45
A.12	Resultados do modelo MLP com dropout de 30% no conjunto de treino.	45
A.13	Resultados do modelo GRU no conjunto de teste.	46
A.14	Resultados do modelo GRU no conjunto de treino.	46
A.15	Resultados do modelo GRU com normalização no conjunto de teste.	47
A.16	Resultados do modelo GRU com normalização no conjunto de treino.	47
A.17	Resultados do modelo GRU com normalização e dropout de 60% no conjunto de teste.	48
A.18	Resultados do modelo GRU com normalização e dropout de 60% no conjunto de treino.	48

A.19 Resultados do modelo GRU com normalização e dropout de 30% no conjunto de teste.	49
A.20 Resultados do modelo GRU com normalização e dropout de 30% no conjunto de treino.	49
A.21 Resultados do modelo GRU com dropout de 60% no conjunto de teste.	50
A.22 Resultados do modelo GRU com dropout de 60% no conjunto de treino.	50
A.23 Resultados do modelo GRU com dropout de 30% no conjunto de teste.	51
A.24 Resultados do modelo GRU com dropout de 30% no conjunto de treino.	51
A.25 Resultados do modelo LSTM com dropout de 30% no conjunto de teste.	52
A.26 Resultados do modelo LSTM com dropout de 30% no conjunto de treino.	52
A.27 Resultados do modelo LSTM no conjunto de teste.	53
A.28 Resultados do modelo LSTM no conjunto de treino.	53
A.29 Resultados do modelo LSTM com normalização no conjunto de teste.	53
A.30 Resultados do modelo LSTM com normalização no conjunto de treino.	54
A.31 Resultados do modelo LSTM com normalização e dropout de 60% no conjunto de teste.	54
A.32 Resultados do modelo LSTM com normalização e dropout de 60% no conjunto de treino.	55
A.33 Resultados do modelo LSTM com normalização e dropout de 30% no conjunto de teste.	55
A.34 Resultados do modelo LSTM com normalização e dropout de 30% no conjunto de treino.	56
A.35 Resultados do modelo LSTM com dropout de 60% no conjunto de teste.	56
A.36 Resultados do modelo LSTM com dropout de 60% no conjunto de treino.	57
A.37 Resultados do modelo Linear no conjunto de treino.	57
A.38 Resultados do modelo Linear no conjunto de teste.	57
B.1 Resultados do modelo MLP no conjunto de treino.	59
B.2 Resultados do modelo MLP no conjunto de teste.	59
B.3 Resultados do modelo LSTM no conjunto de treino.	59
B.4 Resultados do modelo LSTM no conjunto de teste.	59
B.5 Resultados do modelo GRU no conjunto de treino.	60
B.6 Resultados do modelo GRU no conjunto de teste.	60

Capítulo 1

Introdução

Com o grande crescimento da quantidade de dados captados e armazenados e a necessidade do processamento dos mesmos, algumas áreas da computação tiveram grande destaque e têm sido muito importantes para o desenvolvimento de novos produtos e tecnologias. Recentemente, a área de Inteligência Artificial tem se desenvolvido de maneira acelerada e parte do seu crescimento é dado ao grande poder computacional existente. Essa área tem obtido cada vez mais resultados e isso faz com que outras áreas do conhecimento queiram incorporar suas técnicas.

Atualmente grandes empresas possuem computadores com múltiplas placas gráficas para processar dados e modelos cada vez maiores, visando solucionar problemas cada vez mais difíceis.

Na área de Inteligência Artificial existe o Aprendizado de Máquina, o qual já existe há muitos anos e tem se desenvolvido muito nas últimas décadas. O projeto atual utiliza-se do aprendizado de máquina como base, mais especificamente das redes neurais artificiais.

As redes neurais artificiais são inspiradas no cérebro e foram criadas com o objetivo de solucionar problemas. Porém, caíram em desuso pouco após sua criação devido a certas limitações que os modelos mais simples de redes neurais possuem. Depois de muitos anos, apenas na década de 80, elas voltaram a chamar atenção com a criação do algoritmo de retropropagação. Contudo, seu uso decaiu novamente com a criação de outros modelos de aprendizado de máquina como a máquina de vetores de suporte(SVM). No ano de 2006, Hinton introduziu técnicas de treinamento não supervisionadas das redes neurais artificiais que permitiu o ajuste dos parâmetros de redes com mais camadas. Esse ressurgimento das redes neurais gerou um novo termo na área de Inteligência Artificial, o Aprendizado Profundo. O aprendizado profundo já se mostrou de grande importância em diversos problemas, como visão computacional [1], reconhecimento de fala [2], reconhecimento de padrões em texto [3], entre outros.

Mercado Financeiro

O mercado financeiro é, por definição, um ambiente de compra e venda de valores mobiliários, câmbio e mercadorias[4]. O objetivo de qualquer investidor é maximizar seus ganhos. No mercado financeiro, isso ocorre quando você compra e vende suas ações no momento correto. Entretanto, prever se uma ação irá subir ou descer não é uma tarefa trivial. Isso ocorre principalmente pela não estacionariedade da série financeira e diversos fatores externos que afetam o valor da bolsa de valores e suas ações, como notícias e atos políticos, entre outros.

A teoria chamada de caminhada aleatória [5] consiste na afirmação de que o mercado funciona de forma aleatória e por isso não é possível prever os seus movimentos futuros olhando para os seus movimentos passados. Porém, com os avanços tecnológicos atuais e a grande quantidade de informação disponível, alguns modelos mostraram possuir uma performance melhor que um processo aleatório [6][7][8], muitos desses utilizando informações externas a série financeira, como notícias de fontes confiáveis.

1.1 Motivação

A motivação do presente trabalho é verificar a hipótese de que se é possível prever o comportamento de uma série temporal do mercado financeiro com redes neurais, apenas usando seu valor e índices no tempo. Por mais difícil que a tarefa seja, a aplicação de um bom modelo no mercado financeiro pode gerar muitos lucros.

1.2 Proposta e Objetivo

O objetivo e proposta deste trabalho são explorar diversas topologias e técnicas de redes neurais artificiais para uma série temporal do mercado financeiro. Além de verificar se apenas os índices e valores da série financeira no tempo são suficientes para gerar um modelo preditivo com as redes neurais criadas neste trabalho.

1.3 Organização do documento

O documento está organizado da seguinte forma. Na seção 2, consta a revisão bibliográfica, onde serão apresentados os conhecimentos necessários para entendimento do que foi feito e trabalhos relacionados. Na seção 3, são introduzidas as tecnologias utilizadas e os dados que foram extraídos e usados neste trabalho, enquanto que, na seção 4, o procedimento experimental e análise dos resultados. Por fim, a seção 5 que apresenta as conclusões e trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Nesse capítulo, será feita uma revisão e apresentação das arquiteturas e técnicas usadas nesse trabalho. Os tópicos a serem apresentados são as redes neurais, seguido por redes neurais recorrentes, sendo então descritos os diferentes métodos de otimização explorados, apresentados os algoritmos de treinamento das redes neurais e métodos para melhoria de performance, no qual serão abordados a normalização e dropout.

2.1 Redes Neurais Feedforward

Redes neurais artificiais são modelos computacionais inspirados no cérebro humano[9]. Existem diversos tipos de arquiteturas de redes neurais como as feedforward, as convolucionais e recorrentes. Nesse trabalho foram exploradas duas arquiteturas a recorrente e a rede neural feedforward ou perceptron de múltiplas camadas.

As redes neurais feedforward são conhecidas por esse nome, dado que a informação flui somente em um sentido, para frente, ou seja, não existe uma recorrência dentro dessa rede. Essas redes são de extrema importância para o universo de aprendizado de máquina e são base para diversas aplicações comerciais. Para entendermos essa arquitetura um pouco melhor, começaremos com seu modelo mais simples, o de um único neurônio, o perceptron.

2.1.1 Perceptron

O perceptron, proposto por Rosenblatt [10], é um modelo simples de uma rede neural feedforward e pode ser considerado um classificador linear. Ele recebe um vetor de entradas multiplicado por pesos junto com um viés para o cálculo de sua saída, inicialmente sua função de ativação era linear e para classificação era aplicada uma função degrau ao resultado de saída, uma ilustração do perceptron pode ser vista

na Figura 2.1. Assim, um perceptron ou neurônio artificial por ser formulado por:

$$y = \sigma \left(\sum_j W_j x_j + b \right) \quad (2.1)$$

onde j representa os atributos e para cada atributo existe um peso associado.

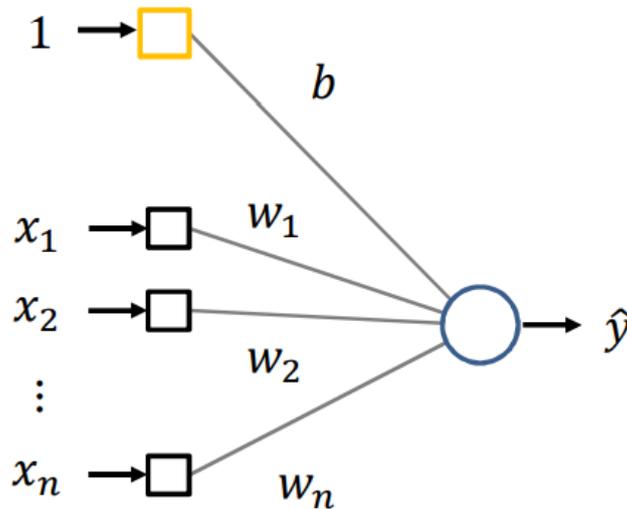


Figura 2.1: Perceptron ou neurônio simples. [11]

O termo $\sum_j W_j x_j + b$ é conhecido como o potencial de ativação. No caso apresentado, está sendo aplicado nesse potencial uma função de ativação sigmoide. Com o uso dessa função, não é mais necessário aplicar a função degrau a saída, basta escolher um limiar para a classificação. Sua saída fica limitada entre os valores 0 e 1, de modo que, conforme o potencial da função cresce, seu valor de ativação se aproxima de 1 e, quando decresce sua ativação, se aproxima de 0. Isso ocorre devida a saturação nos extremos da função. Podemos ver a representação dessa função na Figura 2.2.

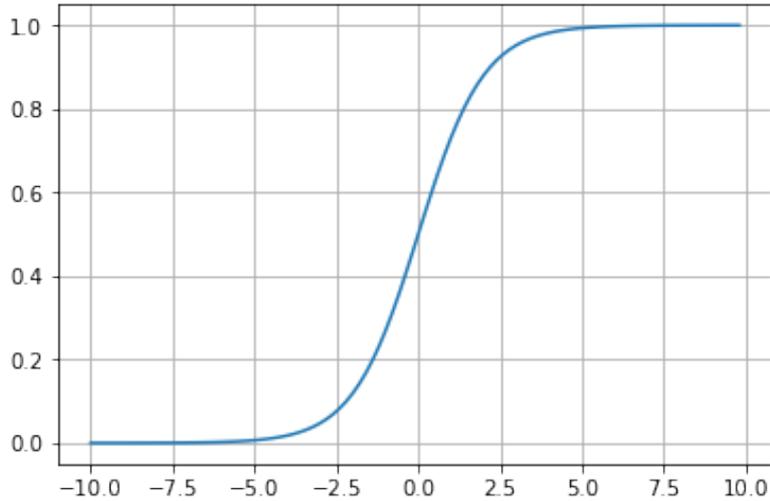


Figura 2.2: Função sigmoíde.

2.1.2 Perceptron de Múltiplas Camadas

O Perceptron de Múltiplas Camadas ou Rede Feedforward Profunda é uma arquitetura de rede neural que consiste em neurônios artificiais em cascata. Onde cada camada é um conjunto de neurônios totalmente conectados através de sinapses na camada seguinte, mas não se conectam entre si, representação visual na Figura 2.3. Em uma rede desse tipo, o número de neurônios por camada escondida é arbitrário. Usando l para expressar a respectiva camada do neurônio, a equação de um único neurônio pode ser representada da seguinte maneira:

$$h^l = \sigma \left(\sum_j W_j^l h_j^{l-1} + b^l \right) \quad (2.2)$$

$$h^1 = \sigma \left(\sum_j W_j^1 x_j + b^1 \right) \quad (2.3)$$

Nessas equações, j representa cada atributo de x em h_1 e cada saída da camada anterior no caso de h_l . h^l é a saída de cada neurônio e h^1 representa o neurônio da primeira camada escondida, a qual recebe o vetor de atributos para calcular sua saída. A partir da segunda camada escondida, é usada a saída da camada anterior como entrada.

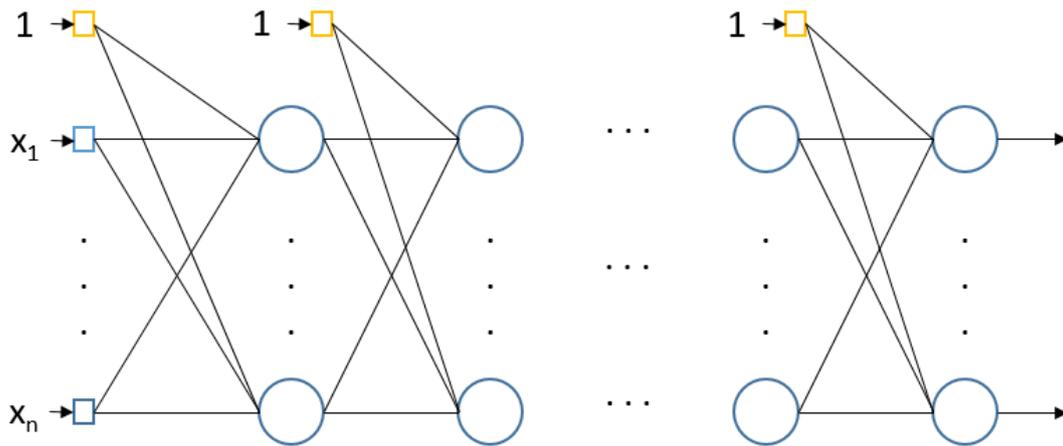


Figura 2.3: Exemplo do Perceptron de Múltiplas Camadas, onde cada camada é totalmente conectada com a seguinte e o número de neurônios por camada é arbitrário.

2.2 Redes Neurais Recorrentes

Redes neurais recorrentes ou RNN são uma família de redes neurais para processar dados sequenciais [12]. É possível ver as redes recorrentes sendo usadas em processamento de linguagem natural, visão computacional e entre outras áreas.

As RNN são uma evolução das redes feedforward e sua principal diferença é o compartilhamento de parâmetros. As redes recorrentes recebem uma entrada assim como as redes feedforward, mas elas também possuem uma recorrência de suas saídas. Nessas redes, cada entrada e cada recorrência possui seus parâmetros e estes são compartilhados em cada passo de tempo. Isso faz com que seja necessário um número muito menor de parâmetros do que em uma rede feedforward com o mesmo número de neurônios e entradas, mas essa redução limita a paralelização do treinamento.

Atualmente, as redes recorrentes mais utilizadas em problemas práticos são as gated RNN. As gated RNN criam caminhos pelo tempo de modo que seja possível mapear as dependências de longo prazo[12]. A ideia dessas redes é conseguir esquecer e usar o passado quando necessário. Dentre as RNN existentes duas foram exploradas nesse trabalho, a LSTM e GRU.

2.2.1 LSTM

A Long-short term memory ou LSTM [13], foi criada em 1997 e introduziu a ideia dos portões de controle da memória, para resolver um problema existente nas redes recorrentes tradicionais, que é a memória de longo prazo. Com o passar dos anos, recebeu algumas modificações e ultimamente tem tomado destaque entre as redes recorrentes, inclusive se tornando o estado da arte em alguns problemas [14][2][15]. Podemos ver a representação de uma célula LSTM na Figura 2.4

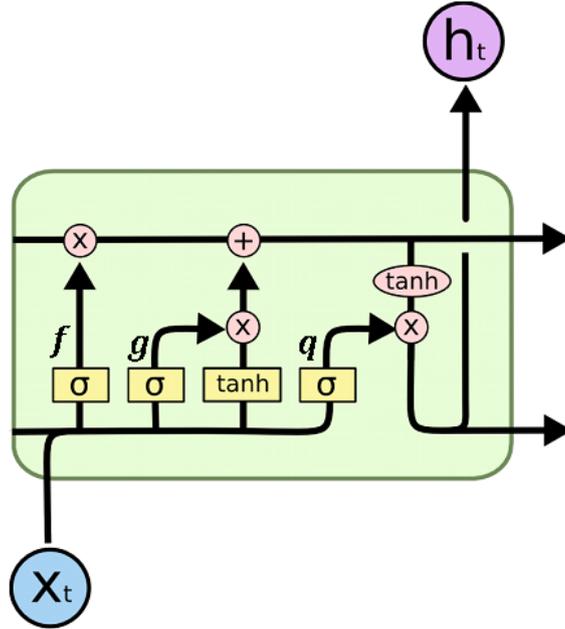


Figura 2.4: Arquitetura de uma célula LSTM. Adaptado de [16]

Em uma célula LSTM, temos as equações dos portões de entradas externas e de esquecimento. Aqui, i representa a célula atual, j cada atributo das amostras ou cada saída da camada anterior, v representa cada célula da camada escondida atual e o fator sobrescrito t representa o instante de tempo:

$$g_i^{(t)} = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_v W_{i,v}^g h_v^{(t-1)} \right) \quad (2.4)$$

$$f_i^{(t)} = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_v W_{i,v}^f h_v^{(t-1)} \right) \quad (2.5)$$

onde x^t é o vetor de entradas atual, $h^{(t-1)}$ é o vetor das saídas das células recorrentes no instante de tempo $t - 1$, b^g , U^g e W^g são, respectivamente, os pesos do bias, os pesos das entradas e os pesos da recorrência do portão de entradas externas e b^f , U^f e W^f são, respectivamente, os pesos do bias, os pesos das entradas e os pesos da recorrência do portão de esquecimento. Com as equações dos portões de

esquecimento e entrada externa, podemos analisar a equação do estado interno da LSTM que é dada por:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \tanh \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_v W_{i,v} h_v^{(t-1)} \right) \quad (2.6)$$

onde podemos ver que existe um termo com uma recorrência com o estado anterior sendo controlada pelo portão do esquecimento e outro que é controlado pelo portão de entradas externas. No termo das entradas externas, temos que b , U e W respectivamente os pesos do bias, os pesos das entradas e os pesos da recorrência da célula LSTM.

A saída $h_i^{(t)}$ da célula LSTM também possui um portão de controle, o portão de saída $q_i^{(t)}$. É possível avaliarmos as equações do portão de saída e da saída respectivamente por:

$$q_i^{(t)} = \sigma \left(b_i^q + \sum_j U_{i,j}^q x_j^{(t)} + \sum_v W_{i,v}^q h_v^{(t-1)} \right) \quad (2.7)$$

$$h_i^{(t)} = \tanh \left(s_i^{(t)} \right) q_i^{(t)} \quad (2.8)$$

tal qual os parâmetros b^q , U^q e W^q são respectivamente os pesos do bias, os pesos das entradas e os pesos da recorrência do portão de saída. Existem diversas variantes dessas equações, por exemplo é possível usar o estado como entrada nos portões.

2.2.2 GRU

A gated recurrent unit ou GRU [17] pode ser vista como uma versão mais simplificada da LSTM, onde serão usados apenas dois portões de controle. Um conhecido como portão de atualização, que faz o papel do portão de esquecimento e de entradas externas ao mesmo tempo, e o outro como portão de reset. Tomando i como representante da célula atual, j representando cada atributo das amostras ou cada saída da camada anterior, v representando cada célula da camada escondida atual e o fator sobrescrito t representa o instante de tempo, suas respectivas equações podem ser vistas a seguir:

$$u_i^{(t)} = \sigma \left(b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_v W_{i,v}^u h_v^{(t)} \right) \quad (2.9)$$

$$r_i^{(t)} = \sigma \left(b_i^r + \sum_j U_{i,j}^r x_j^{(t)} + \sum_v W_{i,v}^r h_v^{(t)} \right) \quad (2.10)$$

A saída $h_i^{(t)}$ da GRU é calculada da seguinte maneira:

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + (1 - u_i^{(t-1)}) \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t-1)} + \sum_v W_{i,v} r_v^{(t-1)} h_v^{(t-1)} \right) \quad (2.11)$$

É possível ver que ambos os portões podem ignorar partes do vetor de estado. Também vemos que é necessário calcular um número menor de parâmetros que a LSTM. Mesmo mais simples a GRU se mostrou eficiente em diversos casos e assim como a LSTM existem algumas derivações dessa arquitetura.

2.3 Funções Custo

Em uma rede neural, assim como em muitos problemas de aprendizado de máquina, uma função deve ser otimizada. Essa função é conhecida como função de custo ou função de avaliação. Cada problema pode ser representado de uma maneira diferente e podem se usar diferentes funções de custos. As mais usadas são o erro médio quadrático e a entropia cruzada. Essas funções são mais utilizadas, pois ao calcularmos a máxima verossimilhança da probabilidade a posteriori chegamos nessas equações. Se assumirmos a função de densidade de probabilidade dos dados como uma distribuição gaussiana chegamos no erro médio quadrático apresentado na seção 2.3.1 e se assumirmos uma distribuição de Bernoulli chegamos a formulação apresentada na seção 2.3.2.

2.3.1 Erro Médio Quadrático

O erro médio quadrático ou *mean squared error* (mse), como o próprio nome deixa explícito, é o cálculo do erro para cada predição do modelo. Podemos verificar sua formulação da seguinte maneira:

$$MSE = \frac{1}{N} \sum_n^N (\tilde{y}_n - y_n)^2 \quad (2.12)$$

onde N é o número de amostras, y_n é o valor correto a ser predito da amostra n e \tilde{y}_n é o valor predito pela rede da amostra n .

2.3.2 Entropia Cruzada

A função de custo entropia cruzada, é calculada a partir da maximização da verossimilhança. A maneira de formular a equação usada nas redes neurais deste trabalho

é a seguinte:

$$L = -\frac{1}{N} \sum_n^N (y_n \log(\tilde{y}_n) + (1 - y_n) \log((1 - \tilde{y}_n))) \quad (2.13)$$

onde N é o número de amostras, y_n é o valor correto a ser predito da amostra n e \tilde{y}_n é o valor predito pela rede da amostra n .

2.4 Métodos de Atualização de Parâmetros

A otimização pode ser definida como uma tarefa de minimizar ou maximizar uma função $f(x)$, alterando-lhe o valor de x [12]. No contexto de redes neurais, é usada para alterar os parâmetros de forma a minimizar ou maximizar uma função de custo, por exemplo, a minimização do erro médio quadrático. Hoje em dia, existem diversos métodos de otimização e alguns foram usados nesse trabalho.

2.4.1 Gradiente Descendente

O gradiente descendente [18] é um método de otimização iterativa para encontrar o mínimo de uma função. Apesar de muito antigo, ainda é usado até hoje em problemas de aprendizado de máquina e muitos outros. Nesse método de otimização, toma-se um passo de acordo com o gradiente da função a ser otimizada, no ponto sendo avaliado. Podemos verificar um caso simples no Algoritmo 1.

Algoritmo 1: Algoritmo do Gradiente Descendente

Entrada: α : Taxa de aprendizado

Entrada: δ : Critério de parada (Valor próximo de zero)

Início

θ_0 : Vetor de parâmetros iniciais

$t \leftarrow 0$ (Inicialização o estante de tempo)

Enquanto θ_t não convergir **faça**

$t \leftarrow t + 1$

 Usando um conjunto de m exemplos do conjunto de treinamento

$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ com seus alvos correspondentes $\mathbf{y}^{(i)}$.

$g_t \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta_{t-1}), \mathbf{y}^{(i)})$ (Cálculo do gradiente)

$\theta_t \leftarrow \theta_{t-1} - \alpha g_t$

Fim

 Parâmetros resultantes θ_t

Fim

Atualmente, existem diversas aplicações do método e uma muito importante que deve ser citada é o gradiente descendente estocástico, o qual aplica o método do gradiente descendente para uma única amostra do conjunto de dados. Uma outra

variante muito usada é o método de mini-batelada ou batelada o qual aplica o método de atualização para um número arbitrário de amostras menor ou igual ao conjunto completo, como exemplificado no algoritmo 1. Esses métodos foram criados com o objetivo de agilizar o processamento do gradiente, visto que a quantidade de dados e de parâmetros dos modelos é tão grande que torna muito alto o custo computacional do treinamento.

2.4.2 Momentum

O método Momentum foi construído com o objetivo de acelerar o aprendizado, especialmente em casos com altas curvaturas, gradientes pequenos mas consistentes ou gradientes ruidosos [19]. Esse algoritmo acumula a média móvel exponencial decadente dos gradientes passados. Podemos analisar o algoritmo do momentum em Algoritmo 2.

Algoritmo 2: Algoritmo do Momentum

Entrada: α : Taxa de aprendizado

Entrada: β : Parâmetro do momento

Início

θ_0 : Vetor de parâmetros iniciais

v_0 : Velocidade Inicial

$t \leftarrow 0$ (Inicialização o estante de tempo)

Enquanto θ_t não convergir **faça**

$t \leftarrow t + 1$

 Usando um conjunto de m exemplos do conjunto de treinamento

$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ com seus alvos correspondentes $\mathbf{y}^{(i)}$.

$g_t \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta_{t-1}), \mathbf{y}^{(i)})$ (Cálculo do gradiente)

$v_t \leftarrow \beta v_{t-1} - \alpha g_t$ (Atualização da velocidade)

$\theta_t \leftarrow \theta_{t-1} + v_t$ (Atualização dos parâmetros)

Fim

 Parâmetros resultantes θ_t

Fim

2.4.3 Adam

A taxa de aprendizado é um dos hiperparâmetros mais difíceis de se escolher e afeta diretamente a performance do modelo. Se supusermos que as direções dos passos estão alinhadas aos eixos, pode ser razoável, usar uma taxa de aprendizado separada para cada parâmetro e adaptável ao longo do treinamento [12]. A partir disso, surgiram algumas heurísticas para mitigar esse problema e uma das primeiras foi o delta-bar-delta [20].

Adaptative Moment Estimation(Adam) [21] é um algoritmo de otimização de taxa de aprendizado adaptável. Ele mantém uma média exponencial dos gradientes e do quadrado dos gradientes. O algoritmo que representa esse tipo de otimização por ser visto em Algoritmo 3. Esse método de otimização se mostra empiricamente melhor que outros métodos adaptativos, como é citado pelos próprios autores.

Algoritmo 3: Algoritmo do Adam

Entrada: α : Tamanho do passo (Valor sugerido: 0.001)

Entrada: $\beta_1, \beta_2 \in [0, 1)$: Taxas de decaimento exponencial para as estimativas de momento (Valores sugeridos: 0.9 e 0.999)

Entrada: δ : Constante pequena para estabilização numérica (Valor sugerido: 10^{-8})

Início

θ_0 : Vetor de parâmetros iniciais

$m_0 \leftarrow 0$ (Inicialização 1º vetor de momento)

$v_0 \leftarrow 0$ (Inicialização 2º vetor de momento)

$t \leftarrow 0$ (Inicialização o estante de tempo)

Enquanto θ_t não convergir **faça**

$t \leftarrow t + 1$

Usando um conjunto de m exemplos do conjunto de treinamento $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ com seus alvos correspondentes $\mathbf{y}^{(i)}$.

$g_t \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta_{t-1}), \mathbf{y}^{(i)})$ (Cálculo do gradiente)

$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ (Atualização da estimativa do primeiro momento enviesado)

$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t \odot g_t$ (Atualização da estimativa do segundo momento enviesado)

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Correção da estimativa do primeiro momento pelo viés)

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Correção da estimativa do segundo momento pelo viés)

$\Delta\theta = -\alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \delta}}$ (Cálculo da atualização)

$\theta_t \leftarrow \theta_{t-1} + \Delta\theta$ (Atualização dos parâmetros)

Fim

Parâmetros resultantes θ_t

Fim

2.5 Algoritmos de Treinamento

Nesta seção, serão apresentados os dois métodos de treinamento de redes neurais artificiais utilizados no trabalho proposto.

2.5.1 Algoritmo de Retropropagação

A dificuldade de treinamento das redes neurais artificiais é um problema recorrente. Em 1986, Rumerhalt, Hinton e Williams propuseram o algoritmo de retropropagação do erro [22], com ele foi possível calcular os pesos de uma rede neural e assim treiná-la para diversos problemas. O algoritmo consiste na retropropagação do erro através das camadas, calculando-se o gradiente da função de custo da rede. Para propagar o erro, usa-se a regra da cadeia para encontrar o valor do gradiente para cada peso a partir do erro obtido pela função custo. Para melhor entendimento do algoritmo de retropropagação, começaremos relembrando a regra da cadeia do cálculo.

Tomando x como um número real e f e g são funções que mapeiam um número real para um número real. Supondo que $y = g(x)$ e $z = f(y) = f(g(x))$, a regra da cadeia pode ser representada por:

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \quad (2.14)$$

Expandindo para vetores, tal qual $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^n$, onde $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ e $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Tomando $\mathbf{y} = g(\mathbf{x})$ e $z = f(\mathbf{y})$, temos representação vetorial na forma:

$$\nabla_{\mathbf{x}} z = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^T \nabla_{\mathbf{y}} z \quad (2.15)$$

onde $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ é a matriz jacobiana $n \times m$ de g . Dessa forma vemos que o gradiente de z em relação a x pode ser obtido multiplicando a matriz jacobiana pelo gradiente de z em relação a y . O algoritmo de retropropagação consiste na aplicação sucessiva da regra da cadeia, a partir da saída e para cada camada em direção as entradas.

Ao se aplicar a regra da cadeia em uma rede neural, assumindo J como a função de custo e para facilitar as notações $\boldsymbol{\theta}$ como representante dos parâmetros W e b , temos:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}; \mathbf{x}; \mathbf{y}) = \left(\frac{\partial \hat{\mathbf{y}}}{\partial \boldsymbol{\theta}} \right)^T \nabla_{\hat{\mathbf{y}}} J(\boldsymbol{\theta}; \mathbf{x}; \mathbf{y}) \quad (2.16)$$

onde $\hat{\mathbf{y}}$ representa o valor estimado pela rede, dessa maneira para computar o gradiente de cada camada basta substituir $\hat{\mathbf{y}}$ por sua equação equivalente e aplicar a regra da cadeia. Melhor detalhamento do procedimento pode ser visto em [12][9][23], uma representação visual do algoritmo pode ser vista em 2.5.

É de extrema importância deixar notório que uma grande vantagem do algoritmo de retropropagação é a sua capacidade de paralelização. Isso ocorre visto que o algoritmo se baseia em multiplicação de matrizes, permitindo, assim, o treinamento de arquiteturas maiores. Hoje em dia, existem redes com milhões de parâmetros treináveis. Porém, esse algoritmo sofre ao adicionarmos mais camadas na rede neu-

ral, visto que o gradiente que chega nas primeiras camadas é muito pequeno fazendo com que não ocorra o treinamento.

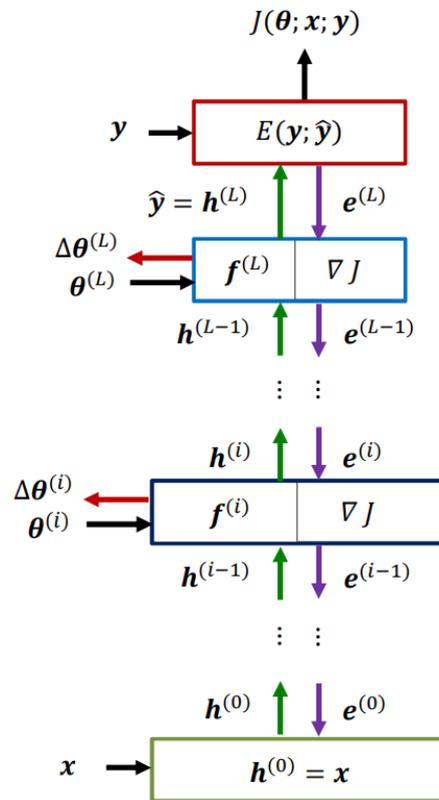


Figura 2.5: Exemplificação visual do algoritmo de backpropagation, onde i representa uma camada intermediária qualquer. [24]

2.5.2 Retropropagação através do Tempo

Retropropagação através do Tempo ou Backpropagation Through Time (BPTT) é nome dado a aplicação do algoritmo de backpropagation em um grafo computacional desdobrado que representa uma rede recorrente, fazendo-se necessário definir uma janela de tempo que será usada [12]. A grande e principal diferença entre o algoritmo de retropropagação aplicado em uma rede neural mais simples e uma rede neural recorrente é que não é possível paralelizar os cálculos na rede recorrente, visto que é necessário usar o estado anterior para calcular o seguinte, uma vez que há o compartilhamento dos pesos. Isso afeta o tempo de execução de um treinamento, tornando-o mais lento em uma rede recorrente, já que ele não pode ser processado em múltiplas cores ou GPU ao mesmo tempo. É possível ver esse algoritmo sendo aplicado nas redes recorrentes citadas nesse trabalho nos livros Supervised Sequence Labelling with Recurrent Neural Networks [14], Deep Learning [12] e no artigo "LSTM: A SearchSpace Odyssey" [25]

2.6 Normalização de Batelada

A normalização de batelada[26] consiste na normalização das saídas de uma camada antes de entrar na camada seguinte, podendo esta ser escalada ou não. A formulação da normalização de batelada da saída de uma camada é dada por:

$$\tilde{h}_j^l = \frac{\gamma^l(h_j^l - \mu^l)}{\sigma^l} + \beta^l \quad (2.17)$$

onde μ^l é a média das saídas da camada l e σ^l é o desvio padrão. Os termos γ^l e β^l são constantes arbitrárias, cujos valores padrão são, respectivamente, 1 e 0. Essas constantes são usadas para o escalamento dos dados.

A normalização das variáveis é feita para manter os valores das saídas com média zero e desvio padrão um. Isso faz com que os valores dos gradientes não explodam para infinito muito facilmente e as funções de ativação não saturem tão rapidamente.

2.7 Dropout

O dropout é uma técnica de regularização[27]. Essa técnica é muito usada em diversos trabalhos, inclusive, pode ser usada junto com outras técnicas de regularização. Uma das suas principais vantagens é o seu baixo custo computacional, ou seja, não consome muito tempo de processamento e memória, quando usada. Outra grande vantagem é a que ela pode ser usada facilmente em quase todos modelos e tipos de treinamento [28][29].

A técnica consiste em, durante o treinamento, aplicar, para cada conexão de entrada de um neurônio, uma multiplicação com uma probabilidade p por 1 e de $1 - p$ de ser multiplicado por 0. A representação matemática para o dropout pode ser vista a seguir.

$$\tilde{h}_j^l = r_j^l * h_j^{(l)} \quad (2.18)$$

onde r_j^l representa a variável de Bernoulli para a camada l e atributo ou parâmetro j . Quando $l = 0$ temos as entradas da rede e para $l \neq 0$ temos as saídas de cada camada anterior, ou seja, apenas uma multiplicação das entradas do neurônio. É importante ressaltar que o termo que zera a entrada do neurônio não é aplicado ao bias. Aplicando 2.18 na equação de saída de um neurônio qualquer de uma rede feedforward, temos:

$$h^l = \sigma \left(\sum_j W_j^l \tilde{h}_j^{(l-1)} + b^l \right) \quad (2.19)$$

com isso, temos o exemplo da equação de um neurônio para aplicação do backpro-

pagation em uma rede feedforward usando dropout.

2.8 Trabalhos Relacionados

A predição de ações e índices da bolsa de valores é um problema muito visado no mundo todo, visto que pode gerar um grande ganho financeiro. Diversas abordagens foram feitas durante os anos e algumas se destacaram, inclusive algumas foram usadas como base para geração deste trabalho.

Existem abordagens que baseiam-se somente em indicadores da série temporal e também usam redes neurais. Mizuno [?] aborda o problema usando uma rede neural totalmente conectada, a qual foi treinada para prever qual operação deve ser feita com o papel, compra, venda ou nada, usando somente índices gerados a partir do valor da ação no tempo, como a média móvel e o desvio da média móvel. Moghaddam [6] também utiliza-se de uma rede neural totalmente conectada para prever o valor do índice NASDAQ no tempo.

Abordagens que apresentaram resultados com maiores taxas de acerto das predições foram as que tentam adicionar informações além dos indicadores, como notícias. Vargas [7] utiliza-se de uma RCNN, que consiste em uma CNN para processar notícias, seguida por uma rede recorrente LSTM que junta as notícias processadas com o índice e indicadores, para tentar prever se o valor do índice subirá ou cairá no dia seguinte. Zhai [8] usa uma topologia de SVMs para processar o texto de notícias e o valor da ação no tempo com alguns índices, chegando a obter 70% de acurácia ao juntar essas informações.

Capítulo 3

Dados e Ferramentas

Em problemas de aprendizado de máquina, principalmente no caso de aprendizado profundo, é necessário um processamento de grandes quantidades de dados. São muitas as ferramentas disponíveis, desde softwares específicos para machine learning como KNIME, Keel e Weka, passando por softwares de cálculos numéricos como Spark, MATLAB®, Scilab e GNU Octave e até mesmo linguagens de programação como Python e R, que dispõem de vasto conteúdo de toolbox gerado pela comunidade de data mining e machine learning.

No capítulo presente, serão apresentadas as tecnologias escolhidas para desenvolvimento, implementação e obtenção dos resultados deste trabalho e serão apresentados os dados utilizados como foram extraídos e pré-processados. Primeiramente, será introduzida a linguagem de programação escolhida e algumas ferramentas baseadas nesta linguagem. Então, serão apresentadas a plataforma para gerenciamento de dependências e a interface utilizada no desenvolvimento. Para concluir a seção de ferramentas, será apresentado o principal framework utilizado no trabalho, o qual é um dos mais famosos e usados nos dias atuais. Na segunda e última seção, serão apresentados os dados, como e de onde foram extraídos, seus pré-processamentos e algumas análises sobre eles.

3.1 Ferramentas

Nesta seção, serão apresentadas as tecnologias utilizadas no desenvolvimento desse trabalho. O projeto consistiu na implementação de redes neurais artificiais para predição de classes de subida e descida em uma série temporal financeira. O processamento de otimização em redes neurais é computacional custoso dependendo do tamanho da rede, a escolha certa de linguagem e ferramentas de desenvolvimento a serem utilizadas é importante. Começaremos a seção apresentando a linguagem na qual o código foi escrito junto com algumas das principais bibliotecas utilizadas para processamento dos dados. Em seguida, serão apresentados o gerenciador de

dependências e a interface na qual o código foi escrito. Concluindo a seção, será apresentada a ferramenta principal para o desenvolvimento do projeto, a qual foi usada para implementação das redes neurais artificiais utilizadas.

3.1.1 Python

Python [30] é uma linguagem de programação de alto nível, interpretada, de script e orientada a objetos. É uma linguagem muito conhecida e difundida, inclusive ganhou o terceiro lugar de linguagem mais amada no stackoverflow [31] em 2018 [32]. A linguagem escolhida foi criada priorizando a legibilidade do código sobre a velocidade de processamento. Devido a essa escolha, usar a linguagem pura, ou seja, sem nenhum framework, não se mostra eficiente para processamentos mais complexos, como, por exemplo, multiplicação de matrizes. Assim, surgiram alguns frameworks para contornar o problema do processamento mais lento.

Numpy

Numpy [33] é um framework de computação numérica muito utilizado e que serve de base para outras bibliotecas, algumas inclusive utilizadas neste projeto. Devido sua implementação diferenciada em C/C++ e FORTRAN, o numpy se mostra muito melhor em processamento de matrizes do que Python puro permitindo trabalhar com arranjos, vetores e matrizes de N dimensões, que são de grande importância para os ajustes de parâmetros em redes neurais. Além disso, mantém a facilidade de uma escrita mais simples e limpa, fornecida pelo Python.

Pandas

Pandas [34] é uma biblioteca open source, de fácil implementação e uso para análise e processamento de dados. É de grande importância nesse projeto, devida a facilidade que provê para visualização e manipulação dos dados.

3.1.2 Anaconda

Um grande problema na área de desenvolvimento é o gerenciamento do ambiente de desenvolvimento. Para mitigar os problemas gerados por um ambiente mal gerenciado, onde as dependências não estão corretamente atualizadas e salvas, foi usado o Anaconda [35]. O Anaconda é um gerenciador de pacotes, ambientes e projetos que facilita o desenvolvimento do trabalho.

Jupyter

O Jupyter [36] é uma aplicação baseada na interface web, que permite o desenvolvimento de software, funcionando como uma IDE. Além das vantagens trazidas por funcionar como uma IDE, outra grande vantagem é sua facilidade de gerar uma documentação limpa e visualmente agradável.

3.1.3 Tensorflow

Existem diversas bibliotecas para desenvolvimento de aplicações de aprendizado de máquina e aprendizado profundo. Uma das principais e com grande destaque por todo o mundo é o Tensorflow [37]. Esse framework é usado amplamente por diversas empresas, como Google, Airbnb, ARM, ebay, Intel, entre outras. O Tensorflow fornece API de alto nível que facilitam a criação e treinamento de modelos, além de fornecer controle de baixo nível para maximizar a flexibilidade e desempenho do sistema.

Em sistemas de grande porte, o desempenho é algo crucial e isso é notório na área de aprendizado de máquina, principalmente em aprendizado profundo, onde existem modelos com milhões de parâmetros, que podem demorar dias para rodar em clusters de computadores. Por isso, o Tensorflow foi criado de modo que seja possível a paralelização dos processamentos em CPU e GPU. Para melhorar mais ainda sua performance, hoje ele inclui o XLA, um compilador de álgebra linear que agiliza a execução do código em processadores e GPU. Porém, esse compilador ainda está em fase de testes e se considera como uma versão alpha[37].

O Tensorflow é uma biblioteca de software de código aberto para computação numérica de alto desempenho. Ela pode ser escrita em Python, Java, Go e C. Funciona em diversos sistemas operacionais, como Linux, Windows e MacOS, havendo inclusive pacotes para aplicações mobile e raspberry.

3.2 Dados

A B3, Brasil Bolsa Balcão, é a bolsa de valores oficial do Brasil. No seu site [38], é possível ter acesso a diversas informações sobre a bolsa de valores, além de diversos produtos e serviços. O acesso aos dados da série histórica da bolsa é gratuito e contém a série temporal de todas as ações comercializadas na bolsa brasileira. É possível fazer download dos dados de ano a ano através do link [39], o qual possui toda documentação necessária para entendimento dos arquivos baixados. Neste projeto, foram utilizados dados da série histórica dos valores da ação do banco do brasil (BBAS3) dos anos de 2007 a 2017.

3.2.1 Extração dos Dados

Para obtenção dos dados utilizados neste trabalho, foi necessário seguir os seguintes passos. Primeiro, os arquivos foram descarregados do site do B3 para os anos desejados. Em seguida, foi necessário concatenar e processar esses arquivos para extrair somente a ação com a qual deseja trabalhar, no caso deste trabalho, a BBAS3. Por último é necessário um pré-processamento para tornar os dados interpretáveis para as bibliotecas utilizadas [39].

Seguindo os passos citados, foi gerado o arquivo utilizado. Esse arquivo possui as seguintes colunas: 'DATA_PREGAO', coluna com todos os dias que possuem dados, 'P_ULTIMO', coluna com o preço de fechamento da ação no dia referenciado, 'P_ABERTURA', coluna com o preço de abertura da ação no dia, 'P_MINIMO', coluna com o preço de mínimo do dia, 'P_MEDIO', coluna com o preço médio do dia, 'P_MAXIMO', coluna com o preço de máximo da ação no dia. Exemplo na Figura 3.1

	P_ULTIMO	P_ABERTURA	P_MINIMO	P_MEDIO	P_MAXIMO
DATA_PREGAO					
2007-06-18	28.40	28.90	28.21	28.45	29.00
2007-06-19	28.20	28.48	28.01	28.21	28.48
2007-06-20	27.95	28.50	27.90	28.25	28.50
2007-06-21	28.40	28.01	27.50	28.06	28.50
2007-06-22	28.30	28.16	28.00	28.20	28.55
2007-06-25	27.90	28.40	27.85	28.09	28.40
2007-06-26	27.94	27.95	27.90	28.05	28.29

Figura 3.1: Exemplo do arquivo de entrada, visualização feita com Pandas

3.2.2 Dados da Ação

A partir do arquivo gerado pela extração dos dados da ação escolhida (BBAS3) foram obtidos 2506 dias de dados. A Figura 3.2 representa o comportamento da série temporal dos preços de fechamento. Nota-se que seu valor oscilou muito ao longo dos anos, chegando a um mínimo de R\$11,41 e ao máximo de R\$37,46. Sua média e desvio padrão no período foram respectivamente R\$24,88 e R\$4,94. Infere-se que a série não é estável e não possui uma recorrência visível, a tornando uma série com alto grau de dificuldade de predição. Contudo ao analisarmos a autocorrelação

dessa série temporal, Figura 3.3, consegue-se inferir que preços dos dias anteriores influenciam no preço do dia atual, desse modo podendo ser usados para previsão de um valor futuro.

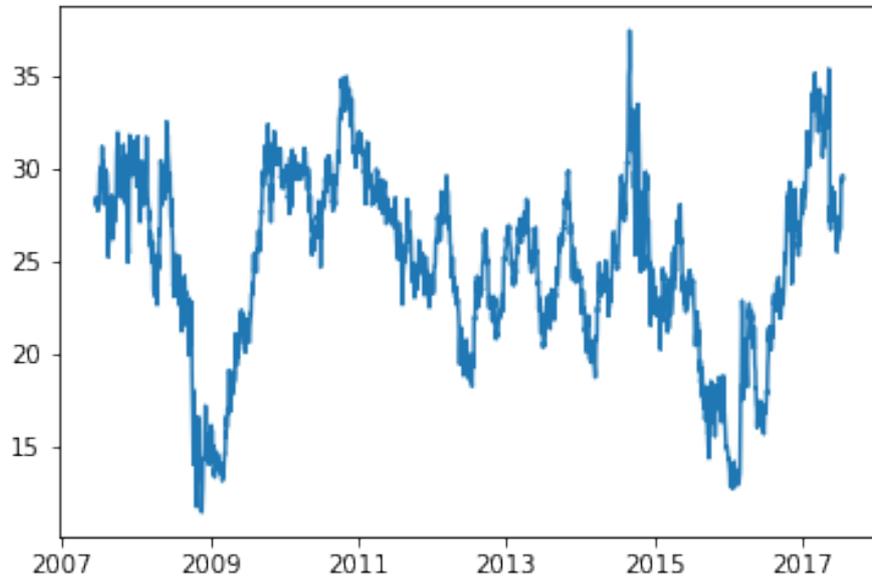


Figura 3.2: Representação visual da série temporal dos preços de fechamento. No eixo X, temos os dias do ano e, no eixo Y, o valor da ação, em Reais.

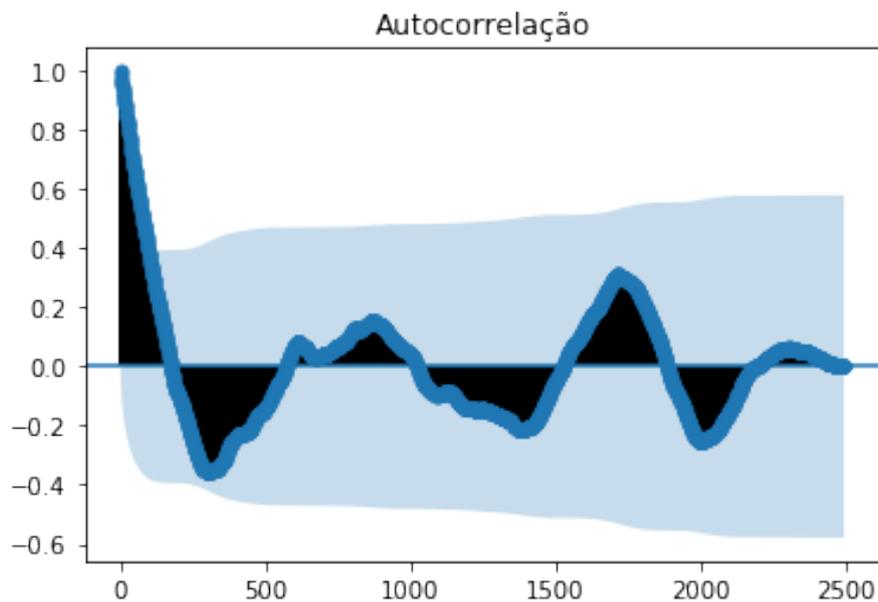


Figura 3.3: Representação visual da autocorrelação dos preços de fechamento da ação BBAS3, o eixo X representa o número de atrasos e o eixo Y a correlação com o atraso 0.

3.2.3 Indicadores

O uso apenas do preço de fechamento de uma ação pode não ser suficiente para prever se o valor da ação subirá ou descerá. Alguns artigos mostraram que certos indicadores podem ser usados para esse tipo de previsão[8][7], sendo estes baseados nos valores de fechamento, mínimo e máximo da série de preços. Para facilitar o detalhamento deles as seguintes siglas serão utilizadas: C_t representa o preço de fechamento no dia t . H_t e L_t são, respectivamente, o valor máximo e mínimo da ação no dia t , HH_5 e LL_5 são o maior máximo e o menor mínimo dos últimos 5 dias, MA_5 é a média móvel do fechamento nos últimos 5 dias. A extensão de dias anteriores usados para os cálculos foi 5, fazendo com que 4 dias fossem perdidos já que eles não possuíam dias anteriores. Os indicadores a seguir foram usados juntamente com o valor de fechamento da série como entrada nos modelos.

1. Indicadores Estocásticos

Indicadores de momento que comparam o preço de fechamento com a segurança do seu preço em um período de tempo.

$$\%K = \frac{C_t - LL_5}{HH_5 - LL_5} \quad (3.1)$$

$$\%D = \frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n} \quad (3.2)$$

2. Indicador de Momento

A taxa de aceleração da segurança do preço da ação, geralmente usado para identificar tendências.

$$Momento = C_t - C_{t-4} \quad (3.3)$$

3. Taxa de Mudança

Representa a velocidade com que a ação muda em um período de tempo.

$$TM = \frac{C_t}{C_{t-4}} \times 100 \quad (3.4)$$

4. Williams %R

O indicador Williams %R é um indicador de momento o qual contabiliza as ações que foram superfaturadas e subfaturadas.

$$\%R = \frac{HH_5 - C_t}{HH_5 - LL_5} \quad (3.5)$$

5. Oscilador A/D

O oscilador acumulação/distribuição (A/D) é um indicador de momento o qual tenta identificar o suprimento e a demanda usando os preços de máximo e mínimo.

$$\text{Oscilador A/D} = \frac{H_t - C_{t-1}}{H_t - L_t} \quad (3.6)$$

6. Disparidade de 5

O indicador de disparidade mede a posição relativa do último preço de fechamento com a média móvel dos últimos dias.

$$\text{Disparidade} = \frac{C_t}{MA_5} \times 100 \quad (3.7)$$

Ao juntarmos os indicadores mostrados com o preço de fechamento da ação temos os atributos de cada amostra apresentada para os modelos. Foi usado um atraso cinco dias para predição, ou seja, para prever o dia $t=6$ usa-se os indicadores e preço de fechamento dos dias $t=[5,4,3,2,1]$. Isso fez com que mais 4 dias fossem perdidos do total de dias obtidos.

Capítulo 4

Análise de Resultados

No presente capítulo serão apresentados o procedimento experimental, detalhando os modelos explorados e a forma como foram treinados, então as figuras de mérito escolhidas e fechando o capítulo será feita a análise de resultados.

4.1 Procedimento Experimental

4.1.1 Modelos

Neste trabalho foram propostos três topologias de redes neurais principais para geração dos modelos, quais sejam: o modelo feedforward e dois modelos recorrentes a LSTM e GRU. De forma que cada modelo gerado deste projeto possui três camadas. No apêndice B, se encontram resultados dos modelos base em uma série temporal mais simples, para mostrar que o código desenvolvido está correto.

A primeira camada é de uma das três topologias, ou seja, ou totalmente conectada ou LSTM ou GRU, seguida de uma camada totalmente conectada e uma terceira camada também totalmente conectada de saída, apresentado na Figura 4.1, onde o primeiro bloco representa as entradas e cada bloco subsequente representa uma camada da rede neural.

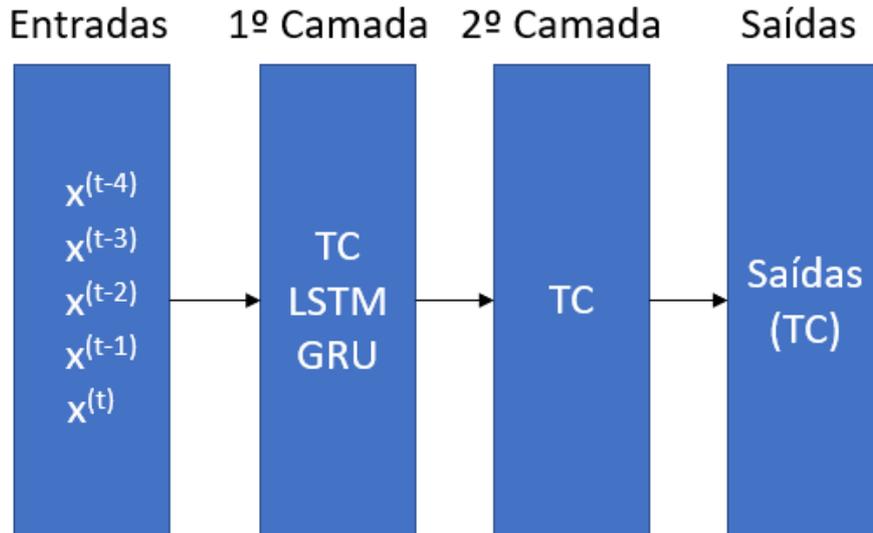


Figura 4.1: Representação visual das topologias escolhidas e usadas. Onde a primeira camada é da topologia a ser testada e as camadas seguintes são totalmente conectadas

Neste sentido, ressalta-se que a escolha da arquitetura foi baseada em alguns modelos que usam uma ou mais camadas totalmente conectadas antes da camada de saída [7][40]. Contudo, poderia ter sido usada somente a camada de saída como totalmente conectada [3] e as outras camadas da topologia a ser testada, como, por exemplo, uma rede com duas camadas recorrentes e uma camada de saída.

Para cada um dos tipos de redes neurais, foram gerados modelos com diferentes números de neurônios em suas camadas. As variações escolhidas para as duas primeiras camadas foram 10 e 10, 25 e 18 e 40 e 30, fazendo com que nas redes recorrentes existam respectivamente 10, 25 e 40 células recorrentes, com cada célula utilizando 5 dias como atraso de tempo. Na terceira camada, a qual é a camada de saída, verifica-se a existência de apenas 2 neurônios, visto que existiam duas classes para serem previstas, as classes "Subirá" e "Descerá", que representam se o valor da ação irá subir ou cair no dia seguinte.

Para cada uma das três variações de neurônios, foram testadas as seguintes técnicas o dropout, a normalização por batelada e três técnicas de otimização.

Para o dropout foram escolhidas 3 variantes, sem dropout, 30% e 60% e o dropout foi aplicado após cada uma das duas camadas escondidas. A normalização por batelada foi aplicada ou não nos modelos, quando aplicada também foi após cada uma das camadas escondidas.

As técnicas de otimização utilizadas foram o gradiente descendente, o momento e adam, gerando assim um total de 18 combinações possíveis de técnicas para cada conjunto de neurônios, ou seja, 54 modelos de redes neurais para cada tipo de rede neural. Logo, foram criados e testados 162 modelos neste trabalho, sendo todo

treinados para otimizar a função de custo da entropia cruzada, Seção 2.3.

Além dos modelos supracitados, foi criado um classificador linear como benchmark de comparação, o qual caracteriza-se por ser um perceptron com função de ativação sigmóide. O qual foi treinado usando o erro médio quadrático, conforme Seção 2.3, para apenas uma classe em vez de duas, ou seja, foi treinado para que 1 representasse a subida do valor da ação e 0 a descida.

4.1.2 Treinamento

Neste trabalho, foi utilizado um treinamento supervisionado dos modelos, ou seja, para todas as entradas do modelo suas saídas eram conhecidas. Como saída foram escolhidas duas classes, a que representa a subida do valor da ação e a que representa a queda desse valor. A formulação dos rótulos seguiu a seguinte regra: se o valor de fechamento da ação do dia seguinte é maior ou igual que o dia atual temos como rótulo a classe de subida, se não temos como rótulo a de descida. Deste modo é necessário saber o fechamento do dia seguinte para gerar os rótulos, perdendo, assim, um dia no conjunto total de dias obtidos através da extração dos dados.

É importante ressaltar que as entradas foram escaladas para que seus valores ficassem entre -1 e 1 antes de serem apresentadas aos modelos, através da equação 4.1, onde X representa todas as amostras de uma certa variável. Ao escalar as entradas, é evitada a priorização de variáveis de entrada que possuem uma escala maior, tornando o treinamento da rede mais estável e evitando saturações rápidas dos neurônios.

$$x_{escalado} = -1 + 2 \frac{x - \min(X)}{\max(X) - \min(X)} \quad (4.1)$$

Os pesos das redes foram inicializados através da heurística criada por Xavier [41], a qual inicializa os pesos com uma distribuição uniforme tendo os limites da distribuição calculados para deixar a escala dos gradientes próximas. Após, era feito um pré-treinamento da rede com as 1300 primeiras amostras com um critério de parada de variação da função custo de 0,1%, critério o qual foi mantido para os treinamentos subsequentes.

Assim, a partir do modelo pré-treinado, as atualizações de peso eram feitas com 30 amostras e testadas com a amostra seguinte. Ou seja, aplica-se uma janela deslizante na série temporal treinando com 30 amostras e testando na amostra seguinte, exemplo na Figura 4.2.

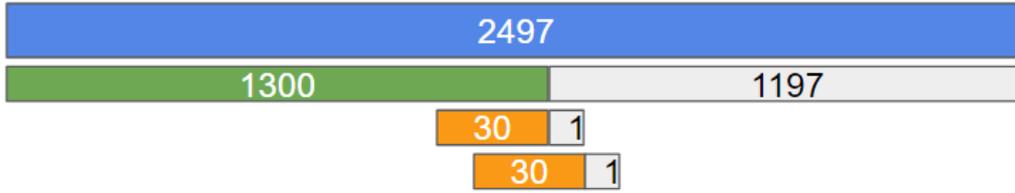


Figura 4.2: Representação visual do treinamento aplicado nas redes neurais criadas.

Existiam 2497 amostras disponíveis para treino, onde 1265 tinham como saída a classe de subida e 1232 de descida. Então, com esse total de amostras, após o pré-treinamento, foi possível testar cada modelo em 1197 dias, dos quais 608 dias são da classe de subida e 588 dias da de descida. Esse treinamento foi inicializado 10 vezes com sementes aleatórias diferentes, assim dando diferentes pontos de partida para as redes e então foi feito uma média e desvio padrão dos resultados para obter um valor demonstrando a estabilidade dos modelos.

4.2 Figuras de mérito

Existem diversas maneiras de se avaliar um algoritmo de aprendizado de máquina [42] [9]. As métricas de avaliação são conhecidas como figuras de mérito, sendo escolhidas para este trabalho acurácia, medida F1 e índice SP, as quais são baseadas na matriz de confusão.

A matriz de confusão é uma matriz que apresenta visualmente a performance de um algoritmo. Podendo ser construída para todas as classes, ou seja, o número de linhas e colunas é o número de classes, ou pode ser feita uma matriz para cada classe do problema, onde é classificado como a classe ou não classe, assim gerando uma matriz de duas linhas e colunas. Para um problema de duas classes, assim como o problema deste trabalho, podemos usar apenas uma matriz de confusão para visualizar e calcular as métricas escolhidas, exemplo visto na Tabela 4.1.

Tabela 4.1: Matriz de Confusão

	\hat{C}_1 (Predita)	\hat{C}_2 (Predita)
C_1 (Real)	VP	FN
C_2 (Real)	FP	VN

Onde VP + FN são todas amostras da C_1 e FP + VN são todas amostras da C_2 . VP é a taxa de verdadeiros positivos, ou seja, as amostras da C_1 que foram preditas como C_1 e são da classe C_1 . FN é a taxa de falsos negativos, ou seja, as amostras da C_1 que foram erroneamente preditas como C_2 . FP é a taxa de falsos positivos, ou seja, as amostras da C_2 que foram erroneamente preditas como C_1 . VN é a taxa

de verdadeiros negativos, ou seja, amostras da C_2 que foram corretamente previstas como C_2 . Com a definição da matriz de confusão é possível calcular as métricas escolhidas.

1. Acurácia

A acurácia consiste em quanto o modelo conseguiu prever corretamente para todas as amostras apresentadas, sua formulação é dada por:

$$Acurácia = \frac{VP + VN}{VP + FN + FP + VN} \quad (4.2)$$

2. Erro

O erro global é calculado pela soma de todas as amostras classificadas erroneamente pelo modelo, sua formulação é dada por:

$$Erro = \frac{FP + FN}{VP + FN + FP + VN} = 1 - Acurácia \quad (4.3)$$

3. Recall

O recall é a quantidade de amostras de uma classe previstas corretamente para todas as amostras daquela classe. A formulação do recall é dada por:

$$Recall(C_1) = \frac{VP}{VP + FN} \quad (4.4)$$

$$Recall(C_2) = \frac{VN}{FP + VN} \quad (4.5)$$

4. Precisão

O precisão é a quantidade de amostras de uma classe previstas corretamente para todas as amostras definidas pelo modelo como daquela classe. A formulação da precisão é dada por:

$$Precisão(C_1) = \frac{VP}{VP + FP} \quad (4.6)$$

$$Precisão(C_2) = \frac{VN}{FN + VN} \quad (4.7)$$

5. Medida F1

A medida F1 [43] é definida como a média harmônica entre a precisão e o recall. Com ela obtemos uma medida para cada classe a ser avaliada e podemos ter

Tabela 4.2: Melhores modelos de acordo com o Índice SP.

Modelo	Acurácia(%)	Índice SP(%)
MLP drop. 60% Mom. [10/10]	52.63 ± 0.13	52.63 ± 0.33
MLP drop. 60% Mom. [25/18]	52.42 ± 0.28	52.41 ± 0.46
MLP drop. 30% Mom. [10/10]	52.38 ± 0.31	52.37 ± 0.83
MLP drop. 60% Mom. [40/30]	52.33 ± 0.27	52.33 ± 0.42
MLP drop. 30% Mom. [25/18]	52.07 ± 0.42	52.06 ± 1.39

uma ideia melhor da performance do algoritmo.

$$F_1 = 2 \times \frac{Precisão \times Recall}{Precisão + Recall} \quad (4.8)$$

6. Índice SP

O índice SP [44] foi criado de maneira a equilibrar as eficiências das classes. Ao avaliarmos a equação 4.9, vemos que, quando a eficiência de uma classe é baixa, o índice tende a ser um valor pequeno. Quando as eficiências são altas seu valor se aproxima de 1. O uso desse índice facilita a comparação de modelos, visto que independente do número de classes teremos apenas um valor.

$$SP = \sqrt{\left(\frac{\sum_{i=1}^{N_{class}} Recall_i}{N_{class}}\right) \cdot \left(N_{class} \sqrt{\prod_{i=1}^{N_{class}} Recall_i}\right)} \quad (4.9)$$

Na equação apresentada $Recall_i$ representa o recall de cada classe, ou seja, a eficiência de cada classe.

4.3 Análise de Resultados

Devido a grande quantidade de modelos criados e testados neste projeto, a apresentação de todos os resultados de treino e teste será feita no apêndice A. Nesta seção, serão apenas avaliados os melhores resultados obtidos no conjunto de teste.

Os modelos foram ordenados pelo índice SP e os cinco melhores modelos podem ser visto na Tabela 4.2. Os cinco melhores modelos, de acordo com o índice SP, foram modelos totalmente conectados, esses mesmos modelos são os que possuem a maior acurácia entre todos modelos testados. É notório que todos os modelos obtidos estão bem próximos entre si e próximos de 50% de acerto. Porém podemos notar pelo desvio padrão, que demonstra a estabilidade dos modelos, dos melhores modelos que eles são regularmente maiores que esse valor.

Tabela 4.3: Melhores modelos LSTM de acordo com o Índice SP.

Modelo	Acurácia(%)	Índice SP(%)
LSTM Mom. [10/10]	51.06 ± 0.25	51.04 ± 0.64
LSTM Mom. [40/30]	50.93 ± 0.82	50.85 ± 2.05
LSTM Mom. [25/18]	50.89 ± 0.51	50.85 ± 1.67
LSTM drop. 30% Mom. [10/10]	50.81 ± 0.77	50.81 ± 1.84
LSTM drop. 60% Mom. [25/18]	50.70 ± 0.89	50.66 ± 7.60

Tabela 4.4: Melhores modelos GRU de acordo com o Índice SP.

Modelo	Acurácia(%)	Índice SP(%)
GRU drop. 60% Mom. [25/18]	51.28 ± 0.43	51.28 ± 1.73
GRU drop. 30% Mom. [10/10]	51.14 ± 0.58	51.13 ± 1.26
GRU drop. 60% Mom. [10/10]	51.07 ± 0.34	51.07 ± 1.54
GRU drop. 60% Mom. [40/30]	51.10 ± 0.41	51.05 ± 3.83
GRU Mom. [40/30]	50.91 ± 0.96	50.84 ± 6.69

Como todos melhores resultados pertenciam apenas ao modelo totalmente conectado, foram geradas as tabelas dos melhores modelos das outras redes neurais que podem ser visto nas Tabelas 4.3 e 4.4.

Para comparação com os melhores modelos e uso como benchmark do projeto os resultados do modelo linear são apresentados na Tabela 4.5. Vemos que seus resultados ficaram na maioria menores que 50%, porém quando o gradiente descendente foi usado como otimizador o resultado conseguiu superar essa marca.

O otimizador Momentum se destacou, visto que ficou presente entre os melhores globais e os melhores de cada tipo de rede. O gradiente descendente só se mostrou melhor para o caso do modelo linear. A técnica de dropout também fez parte da maioria dos resultados, já a de normalização não apareceu em nenhum dos melhores.

Nota-se também que os modelos de redes neurais na maioria ganharam do modelo linear em relação a acurácia e ao índice SP. Porém, não é possível averiguar um ganho significativo do uso das técnicas de redes neurais em relação ao modelo linear, isso provavelmente ocorreu devido a não estacionaridade da série.

Analisando os resultados dos modelos por números de neurônios na camada escondida, Figura 4.3, onde vemos os melhores resultados obtidos com cada conjunto

Tabela 4.5: Resultados do modelo Linear no conjunto de teste.

Modelo	Acurácia(%)	Índice SP(%)
Adam	48.97 ± 0.42	48.97 ± 0.46
Mom.	49.17 ± 0.34	49.16 ± 0.47
G. D.	50.71 ± 1.05	50.56 ± 4.18

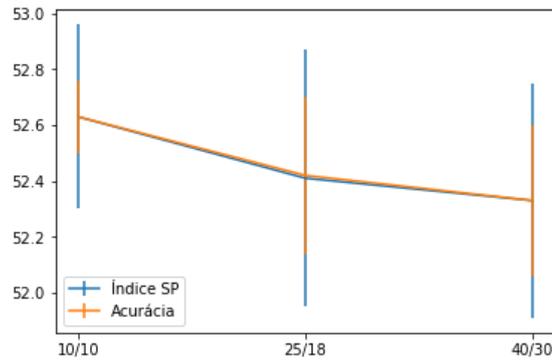


Figura 4.3: Resultados do melhor Índice SP por quantidade de neurônios na camada escondida.

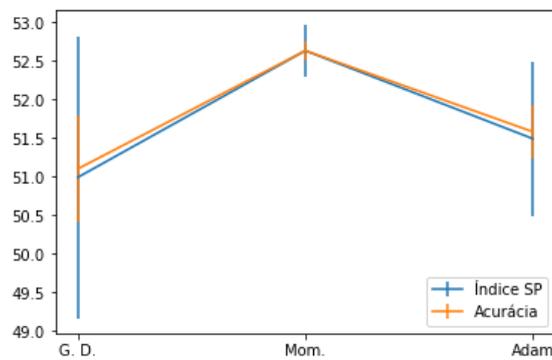


Figura 4.4: Resultados do melhor Índice SP por otimizador.

de neurônios escolhido. Nota-se que os resultados tanto do índice SP quanto da acurácia decaem conforme aumentamos a quantidade de neurônios. Isso pode ter ocorrido pelo fato de uma maior quantidade de neurônios implica em uma maior quantidade de parâmetros necessitando assim de um maior conjunto de treinamento.

Nos resultados dos modelos por otimizador, Figura 4.4, vemos que o melhor modelo obtido otimizador Momentum se mostrou melhor que os outros otimizadores.

Na análise das taxas de dropout usadas, Figura 4.5, nota-se que um aumento da taxa de dropout um modelo com uma taxa de acerto maior. Por último temos os resultados com e sem normalização, Figura 4.6, que mostra que o melhor modelo sem normalização tem uma estabilidade melhor que o modelo com normalização e uma acurácia maior.

Com isso temos todas as análises feitas nesse trabalho e foi possível ver como cada técnica se comportou nos melhores modelos.

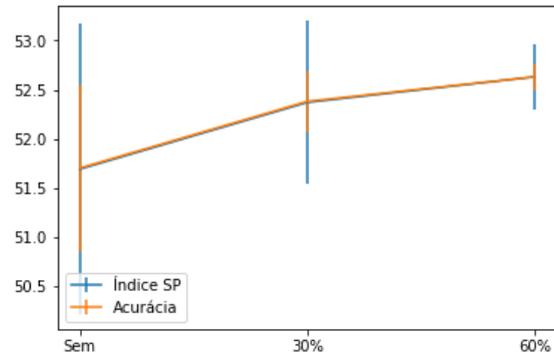


Figura 4.5: Resultados do melhor Índice SP por tipo de dropout utilizado.

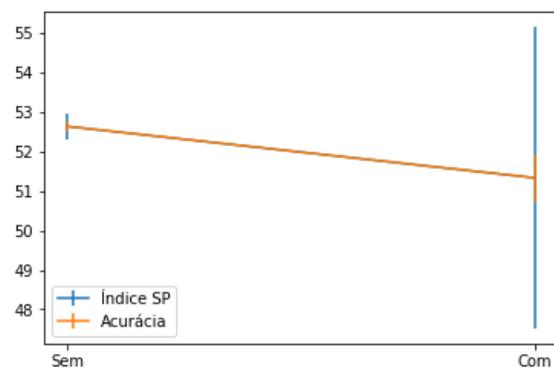


Figura 4.6: Resultados do melhor Índice SP com e sem normalização.

Capítulo 5

Conclusão

Neste capítulo serão feitas a conclusão do trabalho e apresentados possíveis trabalhos futuros.

5.1 Conclusão

Neste trabalho foram vistas algumas abordagens possíveis de redes neurais artificiais para tentar resolver problemas de séries temporais. Como os resultados obtidos foram muito próximos de 50%, não podemos considerar que os modelos obtiveram sucesso na modelagem da série escolhida. A falha no sucesso dos modelos mostra que essa série possui um alto grau de dificuldade. Foi possível concluir que as informações apresentadas aos modelos não foram suficientes.

O principal objetivo do trabalho foi atingido, visto que era desejado explorar diversas topologias e técnicas de melhoria de performance de redes neurais artificiais e verificar seu comportamento em uma série temporal complexa. O total de 162 modelos diferentes foram criados e 1.620 foram treinados e avaliados para obtenção dos resultados, visto que cada modelo foi inicializado com pesos diferentes e treinado 10 vezes.

Através dos resultados obtidos e apresentados, foi possível concluir que os melhores modelos gerados são melhores que os modelos lineares. Contudo essa melhora não é tão significativa, talvez com outras abordagens seja possível alcançar um melhor resultado.

5.2 Trabalhos Futuros

Existem diversas maneiras de evoluir o trabalho corrente. Uma sugestão de abordagem é modificar o número de camadas das redes neurais criadas e remover a camada totalmente conectada intermediária. Assim será adicionado um processamento

hierárquico diferente nas redes, visto que o processamento nas redes recorrentes é diferente das redes totalmente conectadas. Outra abordagem é atacar o problema adicionando informações externas.

Uma opção de adição de informação, que não modifica de forma brusca o código, é adicionar outras ações do mesmo ramo ou relacionadas de alguma maneira com a ação a ser prevista. Uma outra estratégia de adição de informação externa a série temporal que é possível, mas implica na mudança de arquitetura, é o uso de notícias, a qual se mostrou promissora [7][8], tornando uma excelente opção para evolução deste projeto.

Referências Bibliográficas

- [1] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. “ImageNet Classification with Deep Convolutional Neural Networks”, *Commun. ACM*, v. 60, n. 6, pp. 84–90, maio 2017. ISSN: 0001-0782. doi: 10.1145/3065386. Disponível em: <http://doi.acm.org/10.1145/3065386>.
- [2] GRAVES, A., MOHAMED, A., HINTON, G. E. “Speech Recognition with Deep Recurrent Neural Networks”, *CoRR*, v. abs/1303.5778, 2013. Disponível em: <http://arxiv.org/abs/1303.5778>.
- [3] HERMANS, M., SCHRAUWEN, B. “Training and Analysing Deep Recurrent Neural Networks”. In: Burges, C. J. C., Bottou, L., Welling, M., et al. (Eds.), *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pp. 190–198, 2013. Disponível em: <http://papers.nips.cc/paper/5166-training-and-analysing-deep-recurrent-neural-networks.pdf>.
- [4] “BTG Pactual Digital”. . <https://www.btgpactualdigital.com/blog/financas/mercado-financeiro>. Acesso em 2018-09-07.
- [5] MALKIEL, B. G. *A random walk down Wall Street / Burton G. Malkiel*. Norton New York, 1973. ISBN: 0393055000.
- [6] MOGHADDAM, A. H., MOGHADDAM, M. H., ESFANDYARI, M. “Stock market index prediction using artificial neural network”, *Journal of Economics, Finance and Administrative Science*, v. 21, n. 41, pp. 89 – 93, 2016. ISSN: 2077-1886. doi: <https://doi.org/10.1016/j.jefas.2016.07.002>. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2077188616300245>.
- [7] VARGAS, M. R., DE LIMA, B. S. L. P., EVSUKOFF, A. “Deep learning for stock market prediction from financial news articles”, *2017 IEEE International Conference on Computational Intelligence and Virtual Envi-*

ronments for Measurement Systems and Applications (CIVEMSA), pp. 60–65, 2017.

- [8] ZHAI, Y., HSU, A., HALGAMUGE, S. K. “Combining News and Technical Indicators in Daily Stock Price Trends Prediction”. In: Liu, D., Fei, S., Hou, Z., et al. (Eds.), *Advances in Neural Networks – ISNN 2007*, pp. 1087–1096, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN: 978-3-540-72395-0.
- [9] EVSUKOFF, A. “Ensinando Máquinas”. . Material disponibilizado na aula de Inteligência Computacional(COC361) do Centro de Tecnologia da UFRJ.
- [10] ROSENBLATT, F. “The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain”, *Psychological Review*, pp. 65–386, 1958.
- [11] EVSUKOFF, A. “Apresentação Deep Learning - Aula 3”. . Material disponibilizado na aula de Deep Learning(COC891) do Centro de Tecnologia da UFRJ, 2018.
- [12] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [13] HOCHREITER, S., SCHMIDHUBER, J. “Long Short-term Memory”, v. 9, pp. 1735–80, 12 1997.
- [14] GRAVES, A. “Supervised sequence labelling”. In: *Supervised sequence labelling with recurrent neural networks*, Springer, pp. 5–13, 2012.
- [15] SUTSKEVER, I., VINYALS, O., LE, Q. V. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*, pp. 3104–3112, 2014.
- [16] OLAH, C. “Understanding LSTM Networks”. . <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acessado em abril de 2018.
- [17] CHO, K., VAN MERRIENBOER, B., GÜLÇEHRE, Ç., et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, *CoRR*, v. abs/1406.1078, 2014. Disponível em: <http://arxiv.org/abs/1406.1078>.
- [18] GOLDSTEIN, A. A. “Cauchy’s method of minimization”, *Numerische Mathematik*, v. 4, n. 1, pp. 146–150, Dec 1962. ISSN: 0945-3245. doi: 10.1007/BF01386306. Disponível em: <https://doi.org/10.1007/BF01386306>.

- [19] POLYAK, B. T. “Some methods of speeding up the convergence of iteration methods”, *USSR Computational Mathematics and Mathematical Physics*, v. 4, n. 5, pp. 1–17, 1964.
- [20] JACOBS, D. C. “The concept of adversary participation”, *Negotiation Journal*, v. 4, n. 2, pp. 137–142, 1988.
- [21] KINGMA, D. P., BA, J. “Adam: A Method for Stochastic Optimization.” *CoRR*, v. abs/1412.6980, 2014. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14>.
- [22] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. “Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1”. MIT Press, cap. Learning Internal Representations by Error Propagation, pp. 318–362, Cambridge, MA, USA, 1986. ISBN: 0-262-68053-X. Disponível em: <http://dl.acm.org/citation.cfm?id=104279.104293>.
- [23] SATHYANARAYANA, S. “A Gentle Introduction to Backpropagation”. , 2014. Disponível em: https://www.researchgate.net/profile/Shashi_Sathyanarayana/publication/266396438_A_Gentle_Introduction_to_Backpropagation/links/577d124808aeaa6988aba0bc/A-Gentle-Introduction-to-Backpropagation.pdf. Acesso em 2018-09-07.
- [24] EVSUKOFF, A. “Apresentação Deep Learning - Aula 4”. . Material disponibilizado na aula de Deep Learning(COC891) do Centro de Tecnologia da UFRJ, 2018.
- [25] GREFF, K., SRIVASTAVA, R. K., KOUTNÍK, J., et al. “LSTM: A Search Space Odyssey”, *CoRR*, v. abs/1503.04069, 2015. Disponível em: <http://arxiv.org/abs/1503.04069>.
- [26] IOFFE, S., SZEGEDY, C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *arXiv preprint arXiv:1502.03167*, 2015.
- [27] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research*, v. 15, pp. 1929–1958, 2014. Disponível em: <http://jmlr.org/papers/v15/srivastava14a.html>. Acesso em 2018-09-07.
- [28] BAYER, J., OSENDORFER, C. “Learning Stochastic Recurrent Networks”, *ArXiv e-prints*, nov. 2014.

- [29] PASCANU, R., GÜLÇEHRE, Ç., CHO, K., et al. “How to Construct Deep Recurrent Neural Networks”, *CoRR*, v. abs/1312.6026, 2013. Disponível em: <http://arxiv.org/abs/1312.6026>. Acesso em 2018-09-07.
- [30] “Python”. . <https://www.python.org/>. Acesso em 2018-09-07.
- [31] “StackOverflow: Learn, Share, Build”. . <https://stackoverflow.com/>, . Acesso em 2018-09-07.
- [32] “StackOverflow Developer Survey Results 2018”. . <https://insights.stackoverflow.com/survey/2018/#most-loved-dreaded-and-wanted>, . Acesso em 2018-09-07.
- [33] “NumPy: a numerical extension for the computer language Python”. . <https://www.numpy.org/>. Acesso em 2018-09-07.
- [34] “Pandas: Python Data Analysis Library”. . <https://pandas.pydata.org/>. Acesso em 2018-09-07.
- [35] “Anaconda: Manage your packages, dependencies and environments”. . <https://anaconda.org/>. Acesso em 2018-09-07.
- [36] “Jupyter Notebook”. . <http://jupyter.org/>. Acesso em 2018-09-07.
- [37] “Tensorflow: a high performance numerical computation library”. . <http://jupyter.org/https://www.tensorflow.org/>. Acesso em 2018-09-07.
- [38] “Brasil Bolsa Balcão”. . http://www.b3.com.br/pt_br/. Acesso em 2018-09-07.
- [39] “Brasil Bolsa Balcão - Cotações Históricas”. . http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/cotacoes-historicas/. Acesso em 2018-09-07.
- [40] LUAN, S., ZHANG, B., CHEN, C., et al. “Gabor Convolutional Networks”, *CoRR*, v. abs/1705.01450, 2017. Disponível em: <http://arxiv.org/abs/1705.01450>. Acesso em 2018-09-07.
- [41] GLOROT, X., BENGIO, Y. “Understanding the difficulty of training deep feed-forward neural networks”. In: Teh, Y. W., Titterton, M. (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, v. 9, *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. Disponível em: <http://proceedings.mlr.press/v9/glorot10a.html>.

- [42] JAPKOWICZ, N., SHAH, M. *Evaluating Learning Algorithms: A Classification Perspective*. New York, NY, USA, Cambridge University Press, 2014. ISBN: 1107653118, 9781107653115.
- [43] CHINCHOR, N. “MUC-4 Evaluation Metrics”. In: *Proceedings of the 4th Conference on Message Understanding, MUC4 '92*, pp. 22–29, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. ISBN: 1-55860-273-9. doi: 10.3115/1072064.1072067. Disponível em: <https://doi.org/10.3115/1072064.1072067>.
- [44] DOS ANJOS, A., TORRES, R., SEIXAS, J., et al. “Neural triggering system operating on high resolution calorimetry information”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, v. 559, n. 1, pp. 134 – 138, 2006. ISSN: 0168-9002. doi: <https://doi.org/10.1016/j.nima.2005.11.132>. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168900205022552>. Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research.

Apêndice A

Tabelas de Resultados

Neste apêndice se encontram as tabelas com os resultados obtidos para cada modelo criado. A organização é dada pelo tipo de rede neural principal, MLP, GRU e LSTM. Nas tabelas são apresentados os valores obtidos de acurácia, pontuação F1 de cada classe e índice SP.

As linhas de cada tabela representam o método de otimização utilizado e o número de neurônios nas camadas escondidas. Os nomes dos métodos de otimização foram abreviados da seguinte forma, "Mom." representa o momentum e "G.D." o gradiente descendente.

As informações necessárias para referência do modelo que apresentou aqueles resultados se encontram no título de cada tabela.

A.1 MLP

Tabela A.1: Resultados do modelo MLP no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.23 ± 1.30	51.75 ± 1.90	48.63 ± 1.90	50.17 ± 3.64
Adam [25/18]	49.69 ± 1.17	49.81 ± 1.84	49.57 ± 1.71	49.70 ± 3.85
Adam [40/30]	49.72 ± 1.52	49.87 ± 1.53	49.58 ± 1.68	49.73 ± 1.70
Mom. [10/10]	51.70 ± 0.85	52.22 ± 1.08	51.16 ± 1.08	51.69 ± 1.48
Mom. [25/18]	51.43 ± 1.06	51.98 ± 1.22	50.87 ± 1.34	51.42 ± 1.58
Mom. [40/30]	51.11 ± 0.82	52.00 ± 1.17	50.19 ± 1.20	51.09 ± 2.15
G. D. [10/10]	50.88 ± 0.86	51.29 ± 1.35	50.47 ± 1.52	50.88 ± 3.99
G. D. [25/18]	50.28 ± 1.02	52.31 ± 1.24	48.06 ± 1.61	50.16 ± 2.26
G. D. [40/30]	51.10 ± 0.70	53.14 ± 0.96	48.89 ± 1.09	50.99 ± 1.83

Tabela A.2: Resultados do modelo MLP no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	74.72 ± 6.25	74.87 ± 8.11	70.22 ± 10.23	71.87 ± 7.98
Adam [25/18]	89.82 ± 5.55	90.59 ± 4.98	88.11 ± 7.64	88.91 ± 6.45
Adam [40/30]	99.29 ± 0.66	99.32 ± 0.69	99.23 ± 0.68	99.26 ± 0.70
Mom. [10/10]	58.22 ± 1.76	42.03 ± 4.03	40.28 ± 2.94	51.65 ± 2.39
Mom. [25/18]	61.30 ± 2.68	48.20 ± 5.71	46.87 ± 3.34	55.39 ± 3.31
Mom. [40/30]	67.12 ± 3.10	60.03 ± 6.28	57.62 ± 4.95	62.47 ± 4.10
G. D. [10/10]	55.52 ± 1.71	47.68 ± 4.76	45.67 ± 4.17	51.07 ± 3.73
G. D. [25/18]	56.47 ± 0.78	50.47 ± 1.99	43.94 ± 2.25	51.50 ± 1.69
G. D. [40/30]	58.00 ± 0.78	51.38 ± 2.09	45.89 ± 1.90	52.36 ± 1.41

Tabela A.3: Resultados do modelo MLP com normalização no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.24 ± 1.31	50.49 ± 2.78	50.03 ± 2.27	50.25 ± 12.41
Adam [25/18]	51.63 ± 0.53	58.12 ± 2.40	43.46 ± 3.58	50.43 ± 22.93
Adam [40/30]	51.14 ± 0.66	56.82 ± 1.43	43.88 ± 2.43	50.20 ± 18.56
Mom. [10/10]	50.46 ± 1.19	52.73 ± 3.21	48.04 ± 2.64	50.34 ± 20.05
Mom. [25/18]	50.09 ± 1.66	52.85 ± 4.53	47.02 ± 3.56	49.87 ± 20.72
Mom. [40/30]	49.35 ± 1.42	52.10 ± 2.99	46.05 ± 3.87	49.11 ± 20.55
G. D. [10/10]	50.89 ± 0.87	55.74 ± 3.32	41.91 ± 19.96	50.17 ± 29.67
G. D. [25/18]	50.26 ± 0.91	46.63 ± 26.53	47.41 ± 28.29	50.04 ± 38.45
G. D. [40/30]	51.20 ± 0.65	57.89 ± 4.44	39.81 ± 27.59	50.11 ± 34.71

Tabela A.4: Resultados do modelo MLP com normalização no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	54.65 ± 2.09	38.39 ± 9.09	38.66 ± 8.67	50.12 ± 12.29
Adam [25/18]	53.72 ± 2.26	48.71 ± 13.03	28.41 ± 16.74	49.39 ± 23.13
Adam [40/30]	53.65 ± 1.80	46.61 ± 10.75	30.43 ± 14.40	49.73 ± 18.60
Mom. [10/10]	52.37 ± 1.70	42.14 ± 12.31	36.27 ± 12.94	50.41 ± 20.10
Mom. [25/18]	51.10 ± 1.18	44.38 ± 13.22	38.68 ± 14.46	50.76 ± 20.43
Mom. [40/30]	50.74 ± 1.11	45.75 ± 12.89	36.92 ± 12.30	50.73 ± 20.09
G. D. [10/10]	52.22 ± 1.55	46.57 ± 19.61	32.31 ± 21.68	49.90 ± 29.40
G. D. [25/18]	51.68 ± 1.95	40.97 ± 26.94	35.17 ± 24.59	50.28 ± 38.25
G. D. [40/30]	52.58 ± 1.58	49.88 ± 20.10	29.71 ± 25.93	49.83 ± 34.53

Tabela A.5: Resultados do modelo MLP com normalização e dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.33 ± 0.60	51.77 ± 0.99	50.89 ± 1.04	51.33 ± 3.81
Adam [25/18]	50.62 ± 1.10	54.66 ± 1.92	45.81 ± 2.32	50.16 ± 11.06
Adam [40/30]	49.82 ± 1.25	54.16 ± 2.09	44.52 ± 3.17	49.31 ± 10.73
Mom. [10/10]	50.60 ± 1.23	51.84 ± 1.81	49.31 ± 2.11	50.56 ± 4.32
Mom. [25/18]	50.34 ± 0.71	50.43 ± 1.32	50.25 ± 1.21	50.35 ± 8.24
Mom. [40/30]	51.40 ± 1.38	53.05 ± 2.10	49.68 ± 3.03	51.32 ± 10.59
G. D. [10/10]	50.94 ± 0.92	50.74 ± 2.04	51.23 ± 1.79	50.95 ± 13.50
G. D. [25/18]	50.43 ± 0.60	45.33 ± 1.34	54.70 ± 1.05	50.07 ± 14.74
G. D. [40/30]	50.32 ± 0.76	48.72 ± 1.39	51.85 ± 1.30	50.31 ± 10.74

Tabela A.6: Resultados do modelo MLP com normalização e dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	56.47 ± 0.66	37.28 ± 2.79	36.69 ± 2.56	50.15 ± 3.78
Adam [25/18]	53.85 ± 1.71	42.10 ± 6.79	30.99 ± 7.50	50.04 ± 10.72
Adam [40/30]	52.93 ± 1.79	42.33 ± 6.51	30.25 ± 7.46	50.03 ± 10.72
Mom. [10/10]	56.55 ± 1.43	41.07 ± 3.43	38.13 ± 2.93	50.70 ± 4.10
Mom. [25/18]	55.73 ± 1.12	42.41 ± 5.21	41.40 ± 6.17	51.26 ± 7.99
Mom. [40/30]	55.98 ± 1.45	48.29 ± 4.48	43.08 ± 7.37	52.65 ± 10.70
G. D. [10/10]	52.69 ± 0.54	37.69 ± 9.47	39.78 ± 7.72	50.34 ± 13.24
G. D. [25/18]	52.45 ± 0.70	32.18 ± 9.39	45.69 ± 10.33	50.12 ± 14.97
G. D. [40/30]	53.01 ± 0.62	39.07 ± 6.84	44.32 ± 7.67	51.26 ± 10.88

Tabela A.7: Resultados do modelo MLP com normalização e dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.31 ± 0.82	51.69 ± 3.00	51.07 ± 1.36	51.32 ± 12.52
Adam [25/18]	50.12 ± 1.47	53.31 ± 2.63	46.49 ± 3.28	49.85 ± 13.69
Adam [40/30]	50.16 ± 1.14	56.07 ± 1.73	42.41 ± 2.47	49.13 ± 7.96
Mom. [10/10]	50.24 ± 1.04	50.26 ± 2.19	50.22 ± 1.80	50.25 ± 12.36
Mom. [25/18]	50.38 ± 1.00	53.04 ± 1.59	47.43 ± 2.12	50.19 ± 7.43
Mom. [40/30]	49.92 ± 1.28	52.63 ± 2.44	46.88 ± 2.26	49.72 ± 11.54
G. D. [10/10]	50.84 ± 0.76	50.09 ± 3.65	51.66 ± 3.54	50.85 ± 23.25
G. D. [25/18]	50.59 ± 0.59	43.96 ± 3.96	56.01 ± 1.21	49.91 ± 24.91
G. D. [40/30]	50.03 ± 0.98	52.42 ± 1.74	47.25 ± 2.76	49.86 ± 23.21

Tabela A.8: Resultados do modelo MLP com normalização e dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	55.65 ± 0.99	36.52 ± 8.57	36.58 ± 7.89	50.13 ± 12.47
Adam [25/18]	53.41 ± 1.91	40.51 ± 9.19	32.05 ± 9.38	50.11 ± 13.66
Adam [40/30]	52.78 ± 0.82	45.02 ± 5.53	27.57 ± 5.29	49.53 ± 7.76
Mom. [10/10]	54.30 ± 1.64	41.15 ± 9.09	41.87 ± 8.24	50.99 ± 12.32
Mom. [25/18]	54.42 ± 1.60	48.36 ± 6.79	42.39 ± 5.05	52.58 ± 7.20
Mom. [40/30]	54.24 ± 0.95	49.30 ± 6.29	41.45 ± 7.05	52.53 ± 11.34
G. D. [10/10]	52.34 ± 1.07	38.24 ± 17.20	42.82 ± 14.59	50.64 ± 23.22
G. D. [25/18]	52.38 ± 1.17	32.64 ± 17.51	48.93 ± 16.11	50.26 ± 25.03
G. D. [40/30]	53.08 ± 1.07	47.25 ± 13.32	40.39 ± 15.06	51.77 ± 23.56

Tabela A.9: Resultados do modelo MLP com dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.58 ± 0.34	53.33 ± 0.48	49.69 ± 0.53	51.49 ± 1.00
Adam [25/18]	51.34 ± 0.48	53.24 ± 0.62	49.28 ± 0.70	51.23 ± 1.07
Adam [40/30]	51.34 ± 0.86	53.27 ± 1.02	49.23 ± 1.28	51.23 ± 1.69
Mom. [10/10]	52.63 ± 0.13	53.17 ± 0.18	52.09 ± 0.18	52.63 ± 0.33
Mom. [25/18]	52.42 ± 0.28	53.09 ± 0.34	51.73 ± 0.36	52.41 ± 0.46
Mom. [40/30]	52.33 ± 0.27	52.92 ± 0.33	51.73 ± 0.32	52.33 ± 0.42
G. D. [10/10]	50.53 ± 0.78	50.97 ± 1.26	50.11 ± 1.59	50.53 ± 10.15
G. D. [25/18]	50.51 ± 0.68	51.84 ± 1.11	49.17 ± 1.66	50.46 ± 11.90
G. D. [40/30]	50.80 ± 0.62	55.00 ± 1.47	45.83 ± 1.50	50.32 ± 14.25

Tabela A.10: Resultados do modelo MLP com dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	57.82 ± 0.49	45.32 ± 1.29	38.66 ± 1.77	51.22 ± 0.91
Adam [25/18]	59.22 ± 0.88	48.73 ± 2.09	42.14 ± 2.17	52.55 ± 1.35
Adam [40/30]	60.74 ± 0.78	50.16 ± 2.44	47.94 ± 2.73	54.24 ± 1.92
Mom. [10/10]	57.03 ± 0.06	37.21 ± 0.35	36.01 ± 0.31	50.04 ± 0.32
Mom. [25/18]	57.07 ± 0.08	37.38 ± 0.53	36.14 ± 0.39	50.09 ± 0.38
Mom. [40/30]	57.10 ± 0.15	37.76 ± 0.67	36.05 ± 0.29	50.11 ± 0.34
G. D. [10/10]	52.96 ± 0.52	40.68 ± 6.43	40.56 ± 6.66	50.36 ± 10.30
G. D. [25/18]	52.89 ± 0.50	42.64 ± 7.63	39.16 ± 6.77	50.12 ± 11.69
G. D. [40/30]	53.39 ± 0.58	46.43 ± 7.88	35.80 ± 8.61	50.02 ± 14.39

Tabela A.11: Resultados do modelo MLP com dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.17 ± 0.58	52.79 ± 0.92	49.44 ± 1.03	51.10 ± 2.91
Adam [25/18]	51.15 ± 1.09	53.33 ± 1.28	48.75 ± 1.57	51.01 ± 2.02
Adam [40/30]	50.68 ± 0.83	53.82 ± 1.18	47.07 ± 1.49	50.40 ± 2.94
Mom. [10/10]	52.38 ± 0.31	53.12 ± 0.45	51.62 ± 0.46	52.37 ± 0.83
Mom. [25/18]	52.07 ± 0.42	52.70 ± 0.65	51.41 ± 0.65	52.06 ± 1.39
Mom. [40/30]	51.97 ± 0.49	52.54 ± 0.46	51.39 ± 0.65	51.97 ± 0.67
G. D. [10/10]	50.18 ± 0.62	50.71 ± 1.14	49.66 ± 1.15	50.18 ± 9.82
G. D. [25/18]	50.33 ± 0.35	50.92 ± 0.82	49.75 ± 0.62	50.33 ± 7.99
G. D. [40/30]	50.61 ± 0.68	53.18 ± 1.25	47.75 ± 1.62	50.43 ± 8.68

Tabela A.12: Resultados do modelo MLP com dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	60.13 ± 1.36	49.89 ± 3.79	45.05 ± 4.55	53.75 ± 3.02
Adam [25/18]	61.94 ± 1.77	54.13 ± 3.89	46.86 ± 3.43	55.89 ± 2.61
Adam [40/30]	63.71 ± 1.37	59.68 ± 4.71	48.90 ± 4.16	58.01 ± 3.63
Mom. [10/10]	57.09 ± 0.19	37.67 ± 1.00	36.15 ± 0.56	50.14 ± 0.65
Mom. [25/18]	57.41 ± 0.64	39.12 ± 2.19	37.50 ± 0.96	50.49 ± 1.04
Mom. [40/30]	57.57 ± 0.47	40.08 ± 1.25	38.47 ± 1.06	50.72 ± 0.80
G. D. [10/10]	53.49 ± 0.88	44.51 ± 5.80	42.10 ± 7.68	50.51 ± 9.63
G. D. [25/18]	54.11 ± 0.84	45.59 ± 5.28	43.32 ± 4.10	50.57 ± 7.56
G. D. [40/30]	54.78 ± 0.77	47.31 ± 5.75	40.99 ± 5.25	50.74 ± 8.93

A.2 GRU

Tabela A.13: Resultados do modelo GRU no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.44 ± 1.28	52.08 ± 2.27	46.50 ± 2.34	49.25 ± 8.67
Adam [25/18]	50.40 ± 1.16	55.89 ± 1.80	43.34 ± 2.42	49.52 ± 9.82
Adam [40/30]	49.66 ± 1.32	54.91 ± 2.16	42.72 ± 5.09	48.86 ± 20.09
Mom. [10/10]	50.84 ± 0.72	51.94 ± 1.10	49.70 ± 1.18	50.81 ± 2.54
Mom. [25/18]	50.74 ± 1.15	52.14 ± 1.96	49.25 ± 1.61	50.69 ± 4.39
Mom. [40/30]	50.91 ± 0.96	52.45 ± 1.67	49.26 ± 1.61	50.84 ± 6.69
G. D. [10/10]	49.59 ± 1.33	52.32 ± 20.30	37.53 ± 23.65	48.53 ± 30.73
G. D. [25/18]	50.15 ± 0.46	55.87 ± 5.90	43.58 ± 3.45	49.42 ± 25.49
G. D. [40/30]	49.98 ± 0.69	49.27 ± 2.78	50.85 ± 1.35	49.99 ± 22.72

Tabela A.14: Resultados do modelo GRU no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	65.32 ± 3.44	67.57 ± 6.25	63.29 ± 7.87	64.77 ± 9.18
Adam [25/18]	60.55 ± 1.85	65.17 ± 5.03	53.33 ± 8.08	59.53 ± 9.54
Adam [40/30]	58.21 ± 4.71	60.41 ± 17.92	45.08 ± 16.15	56.78 ± 20.87
Mom. [10/10]	53.64 ± 1.18	37.23 ± 3.07	34.93 ± 1.58	50.19 ± 2.39
Mom. [25/18]	54.05 ± 1.42	37.97 ± 4.00	35.03 ± 2.81	50.17 ± 4.50
Mom. [40/30]	55.00 ± 1.06	40.12 ± 3.25	36.93 ± 5.20	50.56 ± 6.67
G. D. [10/10]	50.68 ± 0.59	48.52 ± 19.48	30.37 ± 23.49	49.47 ± 30.87
G. D. [25/18]	51.10 ± 0.60	46.44 ± 16.84	32.26 ± 17.37	49.22 ± 25.07
G. D. [40/30]	51.25 ± 0.48	37.65 ± 15.15	41.06 ± 15.52	49.93 ± 22.41

Tabela A.15: Resultados do modelo GRU com normalização no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.74 ± 1.15	56.30 ± 1.88	40.60 ± 4.61	48.45 ± 19.49
Adam [25/18]	49.03 ± 1.42	50.15 ± 3.01	47.36 ± 5.09	48.99 ± 19.14
Adam [40/30]	49.68 ± 1.51	53.20 ± 3.66	45.56 ± 4.16	49.36 ± 22.71
Mom. [10/10]	49.50 ± 1.37	48.39 ± 2.89	50.47 ± 3.22	49.51 ± 25.92
Mom. [25/18]	49.72 ± 0.82	55.50 ± 1.42	41.65 ± 6.89	48.74 ± 18.61
Mom. [40/30]	49.59 ± 1.66	56.07 ± 2.65	40.69 ± 4.69	48.35 ± 15.26
G. D. [10/10]	49.65 ± 0.93	31.15 ± 28.71	56.54 ± 3.63	48.60 ± 35.82
G. D. [25/18]	49.82 ± 0.94	45.24 ± 27.67	44.78 ± 28.03	49.81 ± 41.54
G. D. [40/30]	50.59 ± 1.01	54.97 ± 21.96	32.91 ± 28.30	48.53 ± 38.64

Tabela A.16: Resultados do modelo GRU com normalização no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.42 ± 1.42	46.32 ± 13.04	27.25 ± 13.43	49.31 ± 19.33
Adam [25/18]	51.70 ± 1.58	40.25 ± 10.96	36.19 ± 12.40	50.63 ± 18.87
Adam [40/30]	51.95 ± 1.11	43.51 ± 14.23	33.19 ± 15.11	50.46 ± 22.83
Mom. [10/10]	51.52 ± 1.16	37.15 ± 16.57	39.76 ± 17.94	50.84 ± 25.49
Mom. [25/18]	51.30 ± 0.94	47.02 ± 11.70	30.77 ± 13.46	50.06 ± 18.46
Mom. [40/30]	51.22 ± 1.51	49.41 ± 11.01	31.13 ± 11.49	50.23 ± 15.09
G. D. [10/10]	49.99 ± 0.90	27.65 ± 27.70	49.11 ± 21.82	49.20 ± 35.71
G. D. [25/18]	50.22 ± 0.54	39.09 ± 28.52	37.25 ± 28.38	50.40 ± 41.44
G. D. [40/30]	51.03 ± 1.20	49.46 ± 25.41	25.44 ± 26.29	48.53 ± 38.57

Tabela A.17: Resultados do modelo GRU com normalização e dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.34 ± 1.07	51.83 ± 2.98	45.84 ± 5.85	49.16 ± 25.03
Adam [25/18]	50.43 ± 1.31	54.57 ± 3.98	45.97 ± 3.83	50.03 ± 20.73
Adam [40/30]	49.98 ± 1.62	56.06 ± 2.20	41.97 ± 3.52	48.87 ± 9.93
Mom. [10/10]	49.78 ± 1.15	50.44 ± 2.12	49.13 ± 2.13	49.78 ± 9.35
Mom. [25/18]	49.27 ± 1.32	41.19 ± 3.74	55.35 ± 2.02	48.45 ± 14.53
Mom. [40/30]	49.76 ± 1.34	50.12 ± 3.83	49.57 ± 4.06	49.75 ± 22.93
G. D. [10/10]	49.62 ± 1.18	44.74 ± 32.01	45.70 ± 26.05	49.55 ± 39.19
G. D. [25/18]	49.77 ± 0.77	40.88 ± 31.11	45.20 ± 26.67	49.75 ± 42.11
G. D. [40/30]	49.67 ± 0.98	33.65 ± 29.93	49.98 ± 26.45	48.21 ± 40.30

Tabela A.18: Resultados do modelo GRU com normalização e dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	52.48 ± 2.30	44.84 ± 13.52	36.01 ± 20.10	51.35 ± 24.99
Adam [25/18]	53.88 ± 1.99	50.96 ± 17.35	40.39 ± 12.62	53.18 ± 20.70
Adam [40/30]	56.96 ± 1.74	60.93 ± 4.58	41.72 ± 9.97	55.59 ± 9.84
Mom. [10/10]	52.90 ± 0.97	39.58 ± 4.80	38.36 ± 8.25	50.82 ± 9.34
Mom. [25/18]	52.61 ± 1.39	31.60 ± 11.71	51.17 ± 9.82	50.65 ± 15.15
Mom. [40/30]	52.04 ± 1.03	41.30 ± 14.83	39.81 ± 17.47	51.35 ± 23.27
G. D. [10/10]	50.10 ± 0.90	36.51 ± 28.32	44.11 ± 25.74	49.86 ± 39.31
G. D. [25/18]	50.20 ± 0.88	36.02 ± 30.97	42.39 ± 26.97	50.22 ± 42.18
G. D. [40/30]	50.09 ± 0.97	27.28 ± 30.52	49.89 ± 26.42	48.96 ± 40.64

Tabela A.19: Resultados do modelo GRU com normalização e dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.25 ± 1.45	55.79 ± 2.69	42.78 ± 5.82	49.33 ± 22.95
Adam [25/18]	50.52 ± 1.39	58.17 ± 1.92	39.32 ± 5.96	48.67 ± 21.36
Adam [40/30]	50.39 ± 1.49	53.63 ± 2.43	46.68 ± 3.18	50.11 ± 9.63
Mom. [10/10]	49.56 ± 1.03	56.00 ± 1.58	40.50 ± 6.46	48.36 ± 20.65
Mom. [25/18]	50.16 ± 1.73	51.76 ± 3.38	48.37 ± 3.90	50.08 ± 18.42
Mom. [40/30]	49.44 ± 1.46	63.51 ± 1.65	17.42 ± 6.49	38.75 ± 10.12
G. D. [10/10]	50.16 ± 1.09	47.44 ± 21.58	47.45 ± 21.14	50.17 ± 36.65
G. D. [25/18]	49.93 ± 0.77	43.75 ± 28.49	51.79 ± 29.44	49.90 ± 42.67
G. D. [40/30]	49.51 ± 1.20	36.96 ± 34.79	49.01 ± 26.50	48.50 ± 40.96

Tabela A.20: Resultados do modelo GRU com normalização e dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	52.47 ± 1.46	47.57 ± 13.99	31.89 ± 15.77	50.56 ± 22.80
Adam [25/18]	54.71 ± 2.23	56.17 ± 13.36	32.47 ± 16.71	51.72 ± 21.79
Adam [40/30]	54.47 ± 1.24	50.13 ± 6.36	41.80 ± 7.01	53.41 ± 9.47
Mom. [10/10]	52.11 ± 1.62	48.35 ± 10.57	28.65 ± 15.44	49.99 ± 20.54
Mom. [25/18]	53.59 ± 2.15	44.78 ± 13.44	41.20 ± 11.75	52.28 ± 18.55
Mom. [40/30]	51.66 ± 1.20	63.31 ± 4.67	12.14 ± 9.13	41.76 ± 10.31
G. D. [10/10]	50.63 ± 1.04	39.84 ± 25.43	42.30 ± 23.59	50.58 ± 36.91
G. D. [25/18]	50.53 ± 1.15	35.95 ± 30.25	42.71 ± 28.08	50.42 ± 42.77
G. D. [40/30]	50.21 ± 0.99	28.81 ± 30.76	49.31 ± 26.66	49.51 ± 41.15

Tabela A.21: Resultados do modelo GRU com dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.18 ± 0.87	51.05 ± 1.48	47.14 ± 1.67	49.08 ± 12.48
Adam [25/18]	49.32 ± 0.90	48.98 ± 2.77	49.56 ± 1.37	49.34 ± 14.61
Adam [40/30]	50.19 ± 1.28	51.16 ± 2.04	49.19 ± 2.20	50.17 ± 5.63
Mom. [10/10]	51.07 ± 0.34	51.40 ± 0.54	50.73 ± 0.55	51.07 ± 1.54
Mom. [25/18]	51.28 ± 0.43	51.60 ± 0.74	50.95 ± 0.69	51.28 ± 1.73
Mom. [40/30]	51.10 ± 0.41	52.33 ± 0.75	49.80 ± 0.78	51.05 ± 3.83
G. D. [10/10]	49.84 ± 0.98	46.08 ± 7.28	53.57 ± 1.87	49.58 ± 28.72
G. D. [25/18]	49.45 ± 0.88	50.72 ± 1.76	47.85 ± 3.38	49.40 ± 17.86
G. D. [40/30]	49.88 ± 0.73	55.21 ± 1.76	43.07 ± 2.31	49.10 ± 25.25

Tabela A.22: Resultados do modelo GRU com dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	60.93 ± 3.79	58.26 ± 10.36	54.58 ± 11.66	58.51 ± 13.59
Adam [25/18]	61.56 ± 4.13	58.45 ± 13.85	56.82 ± 12.39	59.34 ± 15.79
Adam [40/30]	61.51 ± 4.31	56.59 ± 7.21	55.26 ± 8.97	58.25 ± 7.03
Mom. [10/10]	53.16 ± 0.19	35.21 ± 1.01	35.01 ± 1.09	50.02 ± 1.52
Mom. [25/18]	53.30 ± 0.32	35.40 ± 1.24	34.98 ± 0.96	50.03 ± 1.60
Mom. [40/30]	53.47 ± 0.38	36.75 ± 2.70	34.00 ± 2.48	50.01 ± 3.85
G. D. [10/10]	50.32 ± 0.93	37.23 ± 20.93	47.94 ± 18.06	49.91 ± 28.92
G. D. [25/18]	50.59 ± 0.48	42.00 ± 11.39	40.54 ± 13.94	49.55 ± 18.43
G. D. [40/30]	50.86 ± 0.58	46.75 ± 16.14	32.96 ± 19.00	49.38 ± 25.33

Tabela A.23: Resultados do modelo GRU com dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.19 ± 1.67	48.49 ± 2.97	49.82 ± 3.01	49.20 ± 9.84
Adam [25/18]	50.52 ± 0.97	52.83 ± 1.75	47.95 ± 1.83	50.37 ± 9.12
Adam [40/30]	50.40 ± 1.17	51.72 ± 1.85	49.01 ± 2.00	50.35 ± 6.09
Mom. [10/10]	51.14 ± 0.58	51.69 ± 0.74	50.57 ± 0.86	51.13 ± 1.26
Mom. [25/18]	50.63 ± 1.04	51.53 ± 1.68	49.68 ± 1.42	50.60 ± 3.31
Mom. [40/30]	50.46 ± 1.08	52.12 ± 2.26	48.64 ± 1.55	50.38 ± 7.74
G. D. [10/10]	50.06 ± 0.64	46.40 ± 4.10	53.53 ± 1.56	49.84 ± 29.62
G. D. [25/18]	50.07 ± 0.61	50.80 ± 2.55	48.95 ± 5.29	50.06 ± 22.21
G. D. [40/30]	49.90 ± 0.54	54.43 ± 1.05	44.28 ± 2.15	49.34 ± 24.64

Tabela A.24: Resultados do modelo GRU com dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	63.29 ± 5.53	58.44 ± 12.89	61.81 ± 7.84	61.09 ± 10.02
Adam [25/18]	64.33 ± 4.84	65.35 ± 7.44	58.31 ± 11.54	62.07 ± 9.38
Adam [40/30]	67.75 ± 3.39	68.23 ± 5.38	65.19 ± 6.38	65.90 ± 6.84
Mom. [10/10]	53.21 ± 0.40	35.59 ± 0.88	34.69 ± 0.78	50.00 ± 1.05
Mom. [25/18]	53.51 ± 0.88	36.28 ± 2.61	34.54 ± 2.14	50.04 ± 3.24
Mom. [40/30]	53.82 ± 1.37	37.92 ± 5.90	34.66 ± 3.65	50.10 ± 7.48
G. D. [10/10]	50.66 ± 0.61	37.55 ± 21.02	46.64 ± 20.04	49.94 ± 29.92
G. D. [25/18]	50.89 ± 0.37	39.97 ± 12.48	39.75 ± 16.30	49.73 ± 22.41
G. D. [40/30]	51.12 ± 0.47	46.00 ± 15.49	32.97 ± 18.20	49.52 ± 24.66

A.3 LSTM

Tabela A.25: Resultados do modelo LSTM com dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.35 ± 1.73	50.63 ± 2.50	47.99 ± 2.51	49.30 ± 4.59
Adam [25/18]	49.47 ± 1.70	52.66 ± 2.81	45.85 ± 3.34	49.20 ± 14.50
Adam [40/30]	50.08 ± 1.43	51.83 ± 2.27	48.22 ± 2.58	50.00 ± 7.07
Mom. [10/10]	50.81 ± 0.77	51.27 ± 1.04	50.34 ± 1.15	50.81 ± 1.84
Mom. [25/18]	50.28 ± 0.74	52.60 ± 1.21	47.72 ± 1.34	50.12 ± 5.57
Mom. [40/30]	49.25 ± 1.38	48.21 ± 3.26	49.99 ± 3.86	49.26 ± 13.97
G. D. [10/10]	49.46 ± 0.82	47.42 ± 2.48	51.38 ± 1.67	49.42 ± 17.11
G. D. [25/18]	49.31 ± 0.44	49.51 ± 0.85	49.08 ± 1.01	49.31 ± 8.26
G. D. [40/30]	49.54 ± 0.50	48.85 ± 1.07	50.19 ± 1.20	49.55 ± 9.47

Tabela A.26: Resultados do modelo LSTM com dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	72.12 ± 3.25	73.70 ± 4.10	71.15 ± 4.36	71.25 ± 4.99
Adam [25/18]	64.40 ± 6.70	67.42 ± 10.17	58.79 ± 18.21	63.90 ± 14.51
Adam [40/30]	66.11 ± 4.95	66.36 ± 8.38	63.60 ± 10.40	65.16 ± 8.88
Mom. [10/10]	56.64 ± 3.94	41.15 ± 5.20	41.96 ± 7.18	52.84 ± 4.46
Mom. [25/18]	64.51 ± 6.70	59.49 ± 8.47	54.59 ± 14.00	61.35 ± 8.63
Mom. [40/30]	59.80 ± 4.54	51.50 ± 12.73	53.51 ± 13.30	57.71 ± 15.02
G. D. [10/10]	51.33 ± 0.58	33.35 ± 11.49	40.15 ± 11.03	49.69 ± 16.83
G. D. [25/18]	51.45 ± 0.39	39.76 ± 6.29	39.59 ± 5.32	49.62 ± 8.08
G. D. [40/30]	51.66 ± 0.36	37.13 ± 6.59	39.85 ± 5.41	49.74 ± 9.38

Tabela A.27: Resultados do modelo LSTM no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.57 ± 1.11	47.71 ± 2.63	51.23 ± 1.78	49.55 ± 10.52
Adam [25/18]	49.88 ± 1.28	53.25 ± 1.91	46.04 ± 3.86	49.56 ± 16.03
Adam [40/30]	50.38 ± 1.20	62.19 ± 3.27	22.71 ± 24.35	44.01 ± 25.22
Mom. [10/10]	51.06 ± 0.25	51.92 ± 0.35	50.17 ± 0.38	51.04 ± 0.64
Mom. [25/18]	50.89 ± 0.51	52.12 ± 0.73	49.60 ± 0.87	50.85 ± 1.67
Mom. [40/30]	50.93 ± 0.82	52.63 ± 1.08	49.11 ± 1.29	50.85 ± 2.05
G. D. [10/10]	50.08 ± 0.50	54.78 ± 2.64	44.29 ± 3.86	49.54 ± 32.24
G. D. [25/18]	49.88 ± 0.57	55.16 ± 2.08	43.10 ± 4.62	49.14 ± 23.52
G. D. [40/30]	49.60 ± 0.73	51.86 ± 2.82	47.23 ± 1.49	49.48 ± 20.16

Tabela A.28: Resultados do modelo LSTM no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	63.91 ± 3.89	62.44 ± 10.61	65.44 ± 6.50	63.52 ± 10.68
Adam [25/18]	59.90 ± 3.59	64.03 ± 8.75	53.12 ± 15.16	59.24 ± 16.80
Adam [40/30]	54.67 ± 4.83	65.32 ± 12.00	22.18 ± 26.12	49.13 ± 25.16
Mom. [10/10]	53.42 ± 0.21	35.88 ± 0.56	33.97 ± 0.32	50.00 ± 0.64
Mom. [25/18]	53.86 ± 0.40	36.69 ± 1.40	34.01 ± 0.62	50.01 ± 1.56
Mom. [40/30]	54.31 ± 0.65	37.84 ± 1.84	34.35 ± 1.14	50.10 ± 1.81
G. D. [10/10]	50.83 ± 0.64	44.09 ± 20.95	33.01 ± 21.54	49.40 ± 32.11
G. D. [25/18]	50.97 ± 0.47	47.43 ± 11.44	32.49 ± 17.43	49.04 ± 23.76
G. D. [40/30]	51.14 ± 0.28	40.28 ± 14.37	36.98 ± 12.45	49.96 ± 19.86

Tabela A.29: Resultados do modelo LSTM com normalização no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.05 ± 1.35	52.38 ± 8.20	46.13 ± 3.71	49.71 ± 29.29
Adam [25/18]	50.63 ± 1.55	57.91 ± 2.89	40.29 ± 5.61	49.00 ± 19.64
Adam [40/30]	50.08 ± 0.85	56.59 ± 1.38	41.04 ± 2.77	48.76 ± 22.52
Mom. [10/10]	50.28 ± 1.23	49.92 ± 4.40	50.39 ± 3.06	50.29 ± 22.26
Mom. [25/18]	48.92 ± 1.66	51.90 ± 3.38	45.21 ± 3.01	48.66 ± 20.91
Mom. [40/30]	48.85 ± 1.78	48.90 ± 5.58	48.03 ± 5.13	48.84 ± 28.04
G. D. [10/10]	50.27 ± 0.81	52.14 ± 26.69	22.58 ± 29.59	46.99 ± 38.95
G. D. [25/18]	49.82 ± 0.78	52.26 ± 21.83	34.63 ± 27.64	48.90 ± 39.02
G. D. [40/30]	49.85 ± 0.88	53.29 ± 2.39	40.26 ± 25.37	49.48 ± 37.09

Tabela A.30: Resultados do modelo LSTM com normalização no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	52.29 ± 1.65	40.30 ± 20.43	30.88 ± 18.71	49.94 ± 29.30
Adam [25/18]	52.90 ± 1.87	47.88 ± 11.49	25.06 ± 14.16	48.87 ± 19.28
Adam [40/30]	52.15 ± 1.28	45.37 ± 15.31	25.56 ± 15.06	49.09 ± 22.69
Mom. [10/10]	51.91 ± 1.26	36.97 ± 14.98	38.16 ± 15.80	50.46 ± 22.13
Mom. [25/18]	50.56 ± 1.20	41.05 ± 13.81	32.89 ± 14.42	50.24 ± 20.89
Mom. [40/30]	50.81 ± 1.19	38.16 ± 18.40	35.78 ± 18.52	50.32 ± 28.17
G. D. [10/10]	50.38 ± 0.65	50.82 ± 26.43	19.48 ± 26.69	47.15 ± 38.99
G. D. [25/18]	50.44 ± 0.47	46.91 ± 23.94	29.99 ± 28.62	49.90 ± 39.77
G. D. [40/30]	51.12 ± 1.25	44.73 ± 24.60	35.76 ± 26.08	50.69 ± 37.44

Tabela A.31: Resultados do modelo LSTM com normalização e dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.01 ± 0.79	43.23 ± 1.53	55.37 ± 1.33	49.39 ± 15.23
Adam [25/18]	50.59 ± 0.58	58.43 ± 1.40	38.87 ± 10.62	48.65 ± 23.57
Adam [40/30]	50.36 ± 1.35	61.79 ± 2.15	27.96 ± 18.65	44.98 ± 21.28
Mom. [10/10]	49.94 ± 1.08	59.32 ± 2.58	34.42 ± 8.00	46.88 ± 25.72
Mom. [25/18]	50.76 ± 0.64	64.03 ± 1.36	19.96 ± 20.96	42.04 ± 19.81
Mom. [40/30]	50.80 ± 0.43	65.00 ± 1.02	15.84 ± 21.02	39.44 ± 20.29
G. D. [10/10]	49.29 ± 0.62	40.53 ± 6.39	55.87 ± 2.01	48.20 ± 20.28
G. D. [25/18]	49.29 ± 0.97	51.44 ± 3.56	46.93 ± 2.50	49.18 ± 23.73
G. D. [40/30]	49.72 ± 0.97	55.27 ± 3.81	42.33 ± 6.22	48.90 ± 23.28

Tabela A.32: Resultados do modelo LSTM com normalização e dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	53.30 ± 1.76	33.68 ± 12.77	49.59 ± 8.41	51.33 ± 14.85
Adam [25/18]	52.04 ± 0.97	51.48 ± 12.51	27.04 ± 18.24	49.55 ± 23.42
Adam [40/30]	51.24 ± 1.40	56.72 ± 12.03	17.35 ± 15.89	45.79 ± 21.34
Mom. [10/10]	54.39 ± 3.86	57.20 ± 13.54	26.51 ± 24.11	50.06 ± 25.87
Mom. [25/18]	52.47 ± 2.40	59.92 ± 10.90	11.14 ± 15.62	42.30 ± 19.75
Mom. [40/30]	52.18 ± 3.00	63.63 ± 9.19	9.71 ± 18.32	40.31 ± 20.05
G. D. [10/10]	51.22 ± 0.65	26.02 ± 13.37	46.97 ± 14.31	49.03 ± 20.48
G. D. [25/18]	51.55 ± 0.62	40.68 ± 16.13	35.82 ± 15.24	50.62 ± 23.23
G. D. [40/30]	52.05 ± 0.58	47.26 ± 15.05	32.11 ± 14.86	50.33 ± 23.73

Tabela A.33: Resultados do modelo LSTM com normalização e dropout de 30% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	50.03 ± 1.38	54.41 ± 11.18	42.43 ± 26.82	48.98 ± 33.65
Adam [25/18]	50.24 ± 1.15	58.63 ± 1.94	33.11 ± 24.87	47.78 ± 33.32
Adam [40/30]	50.27 ± 1.49	54.12 ± 4.38	43.41 ± 19.49	49.75 ± 26.45
Mom. [10/10]	50.08 ± 1.25	51.02 ± 2.64	49.19 ± 2.47	50.06 ± 17.03
Mom. [25/18]	49.96 ± 1.83	56.57 ± 3.28	39.06 ± 18.58	48.59 ± 25.22
Mom. [40/30]	49.67 ± 1.16	58.70 ± 1.96	34.62 ± 9.51	46.92 ± 20.82
G. D. [10/10]	49.92 ± 0.93	43.96 ± 5.88	54.90 ± 1.27	49.39 ± 19.25
G. D. [25/18]	49.94 ± 0.88	56.60 ± 5.01	40.34 ± 9.95	48.71 ± 33.44
G. D. [40/30]	50.29 ± 0.84	56.86 ± 1.21	39.56 ± 16.33	48.99 ± 18.12

Tabela A.34: Resultados do modelo LSTM com normalização e dropout de 30% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	51.17 ± 1.17	45.27 ± 22.35	27.63 ± 22.72	49.70 ± 33.66
Adam [25/18]	51.32 ± 1.37	50.58 ± 21.18	22.80 ± 23.31	48.61 ± 33.37
Adam [40/30]	51.41 ± 1.28	42.41 ± 18.29	30.18 ± 17.00	50.15 ± 26.56
Mom. [10/10]	54.15 ± 2.41	43.48 ± 12.52	42.34 ± 11.70	52.57 ± 16.99
Mom. [25/18]	53.66 ± 2.30	51.52 ± 16.96	30.75 ± 17.74	51.07 ± 25.55
Mom. [40/30]	52.60 ± 2.05	53.95 ± 11.40	25.56 ± 18.00	49.19 ± 21.30
G. D. [10/10]	52.04 ± 0.73	29.63 ± 14.22	45.37 ± 12.18	49.93 ± 19.82
G. D. [25/18]	51.71 ± 0.83	46.67 ± 21.25	28.28 ± 21.75	49.78 ± 33.01
G. D. [40/30]	53.21 ± 1.04	49.54 ± 10.05	29.54 ± 13.38	50.10 ± 17.78

Tabela A.35: Resultados do modelo LSTM com dropout de 60% no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	49.16 ± 1.29	50.08 ± 3.12	48.06 ± 2.53	49.13 ± 13.08
Adam [25/18]	49.49 ± 0.71	50.57 ± 1.64	48.37 ± 1.48	49.46 ± 8.55
Adam [40/30]	48.57 ± 1.46	50.66 ± 2.42	46.28 ± 2.17	48.45 ± 6.27
Mom. [10/10]	50.65 ± 0.56	51.86 ± 0.92	49.38 ± 0.97	50.61 ± 3.00
Mom. [25/18]	50.70 ± 0.89	51.87 ± 1.29	49.51 ± 1.91	50.66 ± 7.60
Mom. [40/30]	50.43 ± 0.96	51.64 ± 1.63	49.19 ± 1.72	50.39 ± 7.27
G. D. [10/10]	49.46 ± 1.10	47.79 ± 2.78	50.95 ± 1.57	49.45 ± 12.84
G. D. [25/18]	49.28 ± 0.83	52.72 ± 1.38	45.29 ± 1.58	48.96 ± 9.18
G. D. [40/30]	49.63 ± 0.99	48.72 ± 1.91	50.51 ± 1.74	49.64 ± 6.53

Tabela A.36: Resultados do modelo LSTM com dropout de 60% no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	70.98 ± 5.48	72.24 ± 9.22	68.03 ± 9.58	69.30 ± 14.10
Adam [25/18]	70.42 ± 4.69	70.48 ± 10.24	69.40 ± 7.27	69.00 ± 9.67
Adam [40/30]	70.46 ± 3.58	72.41 ± 5.46	69.09 ± 5.30	69.52 ± 6.87
Mom. [10/10]	56.06 ± 2.13	39.38 ± 2.20	37.82 ± 4.81	51.23 ± 2.86
Mom. [25/18]	59.63 ± 5.34	49.92 ± 10.49	47.11 ± 10.55	55.63 ± 9.77
Mom. [40/30]	64.21 ± 6.03	56.94 ± 9.20	55.30 ± 12.94	60.71 ± 9.53
G. D. [10/10]	51.14 ± 0.64	34.22 ± 8.48	38.41 ± 8.36	49.78 ± 12.56
G. D. [25/18]	51.40 ± 0.52	43.08 ± 6.12	34.06 ± 5.59	49.45 ± 8.60
G. D. [40/30]	51.60 ± 0.32	36.19 ± 4.88	39.04 ± 4.65	49.82 ± 6.51

A.4 Linear

Tabela A.37: Resultados do modelo Linear no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam	62.58 ± 0.13	58.83 ± 0.20	57.34 ± 0.21	57.75 ± 0.16
Mom.	60.92 ± 0.10	57.74 ± 0.18	55.85 ± 0.23	56.55 ± 0.16
G. D.	53.16 ± 0.69	50.75 ± 3.82	45.72 ± 4.35	50.11 ± 4.15

Tabela A.38: Resultados do modelo Linear no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam	48.97 ± 0.42	49.40 ± 0.39	48.53 ± 0.48	48.97 ± 0.46
Mom.	49.17 ± 0.34	49.80 ± 0.35	48.53 ± 0.44	49.16 ± 0.47
G. D.	50.71 ± 1.05	53.06 ± 1.58	48.12 ± 1.92	50.56 ± 4.18

Apêndice B

Modelos aplicados a série solar

Neste capítulo são apresentados os resultados de 3 dos modelos utilizados em uma série temporal solar, que pode ser vista na Figura B.1.

Os modelos aplicados foram os básicos, o modelo MLP, LSTM e GRU, todos sem dropout e sem normalização. Todos usaram adam como otimizador. Foram treinados da mesma forma que os modelos para série financeira só que usando 10 dias de atraso de tempo, ou seja, foram usados 10 dias como entrada nos modelos. A janela de inicialização dos pesos foi com 120 amostras e a janela de atualização de pesos foi com 3 amostras. Assim como na série financeira os modelos gerados tentaram prever a subida ou descida do próximo instante de tempo.

Nota-se que essa é uma série mais simples, visto que todos modelos conseguiram uma média de acerto maior que 60% no conjunto de teste.

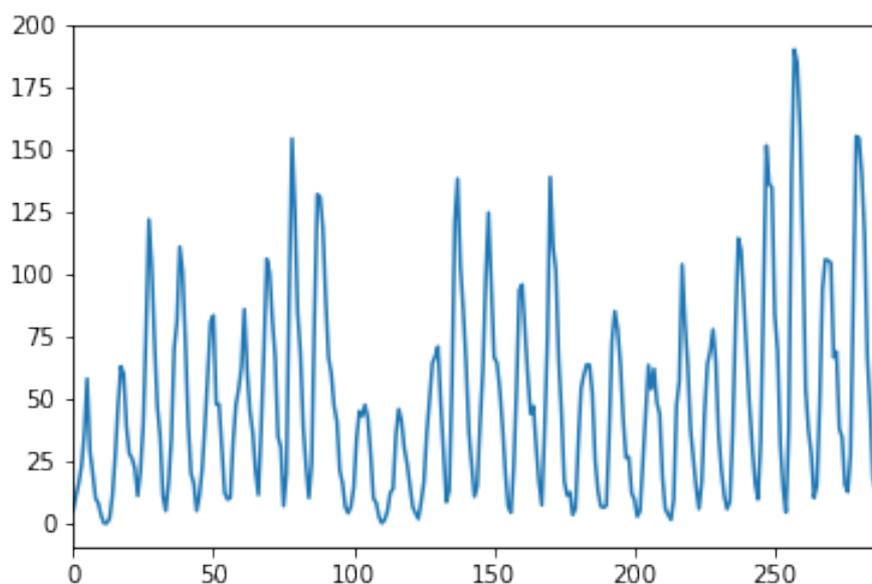


Figura B.1: Representação visual da série temporal solar

B.1 MLP

Tabela B.1: Resultados do modelo MLP no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	98.67 ± 1.26	60.88 ± 2.59	81.05 ± 0.92	70.82 ± 1.51
Adam [25/18]	99.85 ± 0.21	63.14 ± 0.25	82.14 ± 0.24	72.35 ± 0.25
Adam [40/30]	99.94 ± 0.10	63.24 ± 0.00	82.23 ± 0.00	72.43 ± 0.00

Tabela B.2: Resultados do modelo MLP no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	77.59 ± 2.37	72.22 ± 3.28	81.25 ± 2.20	76.97 ± 2.92
Adam [25/18]	78.16 ± 1.50	72.77 ± 2.01	81.78 ± 1.36	77.46 ± 1.75
Adam [40/30]	76.84 ± 1.68	70.99 ± 2.61	80.74 ± 1.77	75.95 ± 2.51

B.2 LSTM

Tabela B.3: Resultados do modelo LSTM no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	84.79 ± 7.81	46.39 ± 8.16	66.44 ± 6.35	58.16 ± 6.69
Adam [25/18]	92.64 ± 2.46	51.90 ± 3.21	75.38 ± 1.71	64.25 ± 2.26
Adam [40/30]	90.84 ± 2.02	50.38 ± 3.69	74.14 ± 3.45	63.01 ± 4.17

Tabela B.4: Resultados do modelo LSTM no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	61.84 ± 7.63	51.98 ± 10.63	68.39 ± 6.86	59.95 ± 9.32
Adam [25/18]	76.96 ± 3.42	70.71 ± 4.99	81.06 ± 3.02	75.82 ± 4.37
Adam [40/30]	77.34 ± 4.14	70.80 ± 5.79	81.64 ± 3.73	75.65 ± 5.48

B.3 GRU

Tabela B.5: Resultados do modelo GRU no conjunto de treino.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	79.75 ± 10.13	33.32 ± 16.00	69.61 ± 3.59	52.56 ± 7.74
Adam [25/18]	87.74 ± 5.12	44.19 ± 9.47	74.16 ± 3.37	59.91 ± 6.22
Adam [40/30]	81.65 ± 8.31	38.53 ± 10.54	69.13 ± 7.18	55.23 ± 8.63

Tabela B.6: Resultados do modelo GRU no conjunto de teste.

Modelo	Acurácia(%)	F1 Subida(%)	F1 Descida(%)	Índice SP(%)
Adam [10/10]	61.46 ± 8.17	39.80 ± 19.77	71.75 ± 6.36	54.73 ± 12.48
Adam [25/18]	74.43 ± 6.91	65.07 ± 11.23	80.21 ± 7.16	71.54 ± 11.06
Adam [40/30]	72.09 ± 8.02	61.34 ± 12.80	78.50 ± 7.99	68.63 ± 12.17